



Localina Recommender Client für das iPhone: Go-Live Vorbereitung

Bachelorarbeit

Abteilung Informatik

Hochschule für Technik Rapperswil

Frühjahrssemester 2010

Autor:	Simon Morger
Betreuer:	Prof. Dr. Markus Stolze
Projektpartner:	Astina AG, Zürich
Experte:	Patrick Steiger
Gegenleser:	Prof. Dr. Lothar Müller

Inhaltsverzeichnis

Abbildungsverzeichnis

Abstract

Die Astina AG entwickelt zusammen mit der Universitat Zürich das Recommender System Localina. Ziel dieser Zusammenarbeit ist die Entwicklung eines Ontologie-basierten Empfehlungssystems für Restaurants. Im Herbst 2009 wurde im Rahmen einer Bachelorarbeit ein erster Prototyp einer nativen iPhone App für Localina entwickelt. Die Benutzerfreundlichkeit und Wartbarkeit dieses Prototypen soll nun in der aktuellen Arbeit verbessert werden. Im Gegensatz zu herkömmlichen Restaurant-Emfehlungssystemen basieren die Empfehlungen von Localina auf dem persönlichen Geschmack des Benutzers. Diesen lernt das Empfehlungssystem kennen, indem der Benutzer selbst Restaurants bewertet. Eine besondere Anforderung an die Applikation ist daher, dass dem Benutzer klar machen wird, dass er im Gegensatz zu herkömmlichen Empfehlungssystemen genügend Restaurants bewerten muss, um gute Empfehlungen zu erhalten.

In einer ersten Projektphase wurden die Probleme des bestehenden Prototypen bezüglich der Benutzerfreundlichkeit anhand von Usability-Tests analysiert. In der Folge wurden verschiedene Konzepte zur Verbesserung des Prototypen entworfen und wiederum anhand von Usability-Tests validiert. Das erarbeitete Konzept floss danach in die Überarbeitung der Localina iPhone App ein. Der Code wurde einem Refactoring unterzogen, um die Maintainability der Codebasis zu verbessern. Mittels Code-Analyse-Tools wurde zudem sichergestellt, dass die Applikation keine Memory-Leaks enthält.

Das Bedienkonzept des bestehenden Prototypen wurde grundlegend überarbeitet. Dabei wurde darauf geachtet, dass der Benutzer bei der Bedienung klarer geführt wird. Um den Benutzer zum Bewerten von Restaurants zu animieren, merkt sich Localina neu seine häufigsten Standorte und schlägt ihm dann Restaurants zum Bewerten vor, die sich in

deren Umkreis befinden. Die Darstellung der Empfehlungen wurde ebenfalls verbessert. Der Benutzer kann nun zwischen personalisierten Empfehlungen und solchen, die auf dem Durchschnittsgeschmack basieren, wählen. Zudem konnte die Reaktionsgeschwindigkeit des Benutzerinterfaces stark verbessert werden, was sich ebenfalls positiv auf die Usability auswirkt. Tests mittels Codeanalyse-Tools zeigen, dass die Codequalität der Applikation den Anforderungen genügt, um im App-Store veröffentlicht zu werden. Dank Verbesserungen an der Codebasis ist die Applikation zudem besser wart- und ausbaubar.

I Technischer Bericht

1 Einleitung

1.1 Ausgangslage

Die Firma Astina AG mit Sitz in Zürich ist ein IT Dienstleister, der Software Lösungen für die Bereiche Information Retrieval (Spidering), Data Processing und Information Filtering entwickelt. Die Astina AG entwickelte zusammen mit der Universität Zürich das Empfehlungssystems Localina. Astina möchte die Technologie interessierten Kunden anhand eines anschaulichen Showcases auf dem iPhone präsentieren können.

1.2 Localina

Localina ist ein Kollaborationsprojekt der Astina AG mit der Förderagentur für Innovation des Bundes (KTI) und der Universität Zürich. Das Ziel des Projekts ist es, eine Empfehlungsplattform für Lokalitäten wie Restaurants, Clubs, Bars und Hotels zu lancieren. Die Universität Zürich entwickelt dabei im Rahmen einer Forschungsarbeit den Algorithmus, während die Astina AG für die Entwicklung der Plattform verantwortlich ist.

1.2.a Wie Localina funtioniert

Die Idee von Localina ist, dass der Benutzer Lokalitäten bewerten kann und im Gegenzug personalisierte Empfehlungen erhält. Dadurch unterscheidet sich Localina von den meisten Plattformen mit Restaurantempfehlungen, deren Empfehlungen meistens nur auf dem Durchschnittsrating der Benutzer basiert.

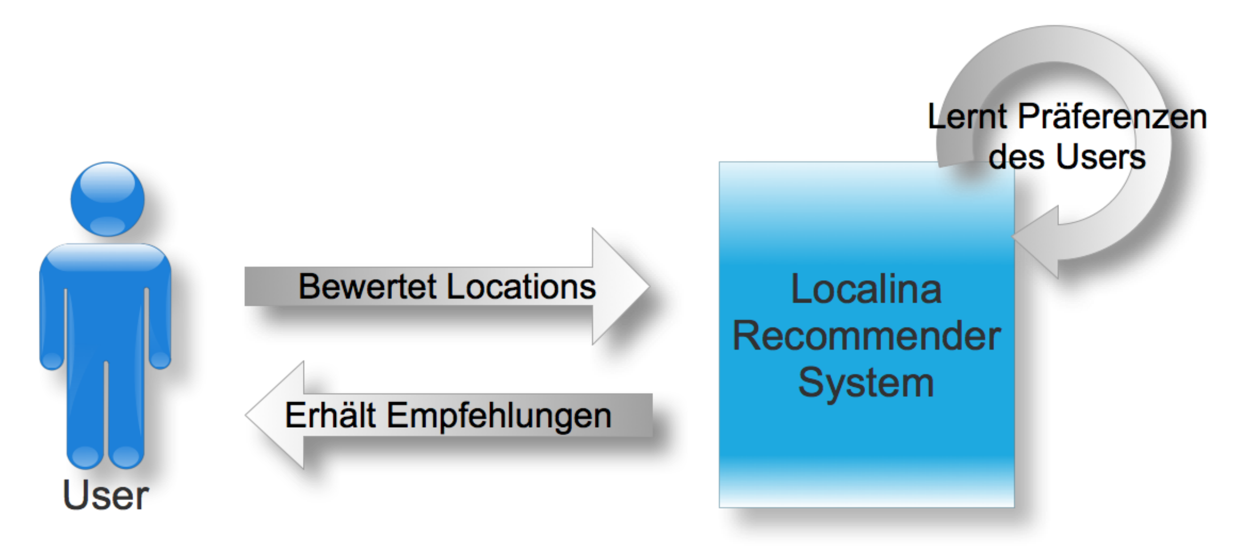


Abbildung I-1: Wie Localina funktioniert

Damit Localina qualitativ gute Empfehlungen liefern kann, muss der Benutzer zuerst genügend Bewertungen abgeben. Das Localina Recommender System kann daraus den persönlichen Geschmack des Benutzers ableiten, indem es versucht, Präferenzen für bestimmte Attribute oder Übereinstimmungen des Bewertungsmusters mit demjenigen anderer Benutzer zu eruieren. Localina implementiert eine Sammlung unterschiedlicher Empfehlungsalgorithmen und entscheidet aufgrund der Datenbasis, auf welche Algorithmen es bei der Prediction zurückgreift. Die Prediction beschreibt die Vorhersage des Recommenders, wie der Benutzer eine Lokalität bewerten würde, sofern er sie kennt, woraus sich folglich die Empfehlung ableitet. Wie in *Abbildung I-2* dargestellt, lassen sich zwei unterschiedliche Ansätze (Strategien) unterscheiden: den Content-based Approach und den Collaborative-Filtering Approach. Beim Content-based Approach werden die Empfehlungen aus den Charakteristiken der Lokalitäten abgeleitet, wogegen beim Collaborative-Filtering Approach die Präferenzen verschiedener User untereinander verglichen werden, um daraus Rückschlüsse auf den Geschmack des Users abzuleiten.

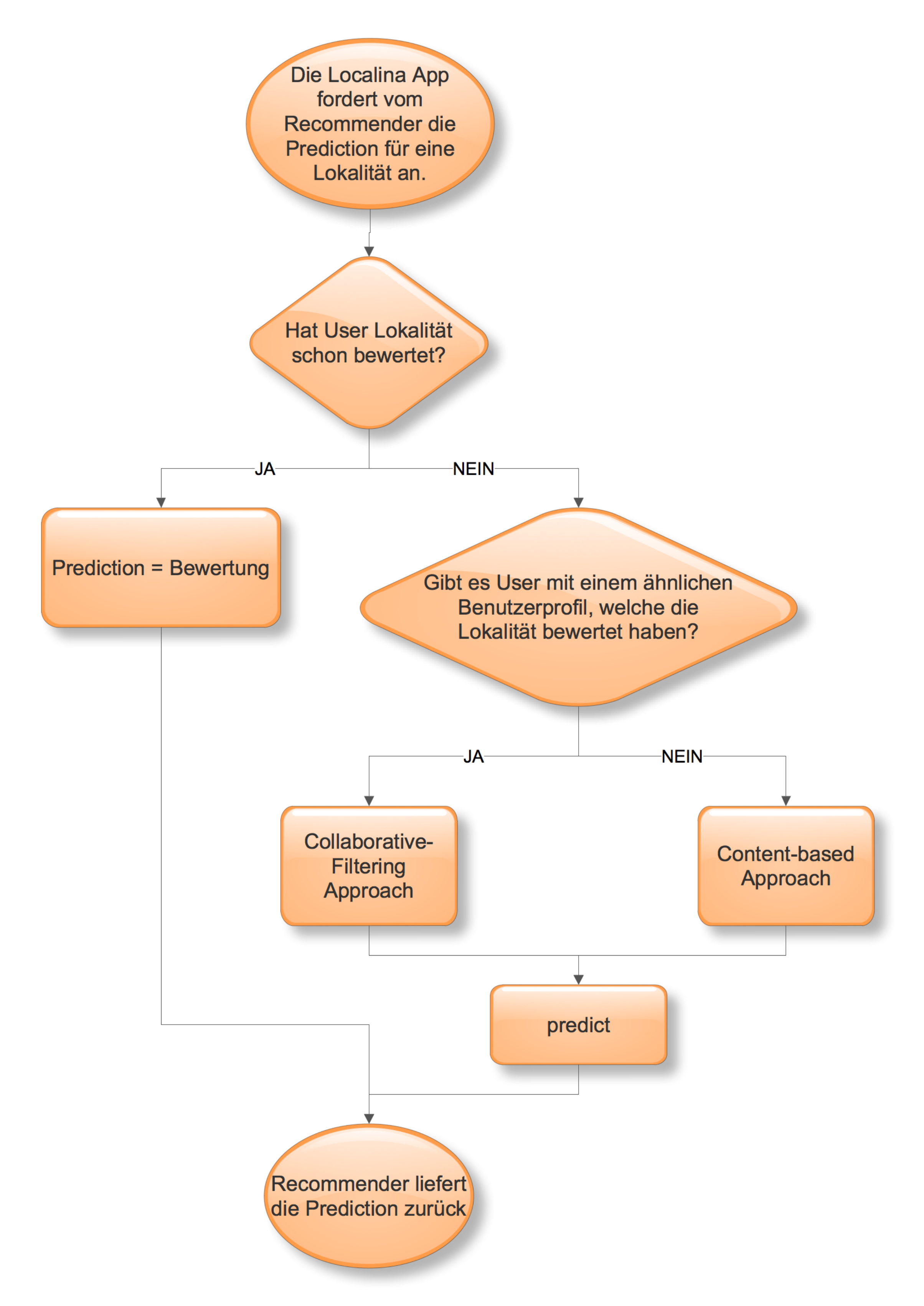


Abbildung I-2: Strategien des Recommenders

1.2.b Aufbau von Localina

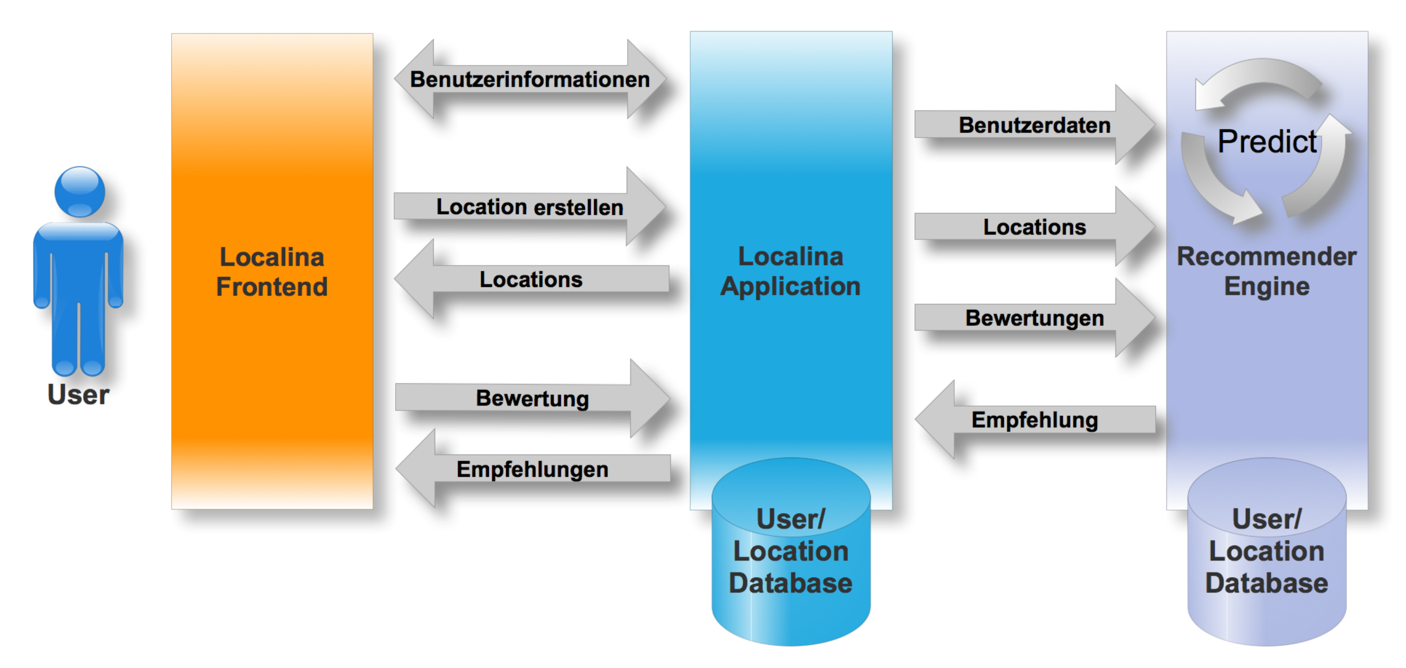


Abbildung I-3: Komponenten von Localina

Das Empfehlungssystem Localina besteht aus folgenden Komponenten:

- Localina Frontend
- Localina Application
- Recommender Engine

Die *Abbildung I-3* stellt die Komponenten und deren Funktionen dar, welche nachfolgend kurz erläutert werden.

Die **Recommender Engine** bildet den eigentlichen Kern des Empehlungssystems. Die Recommender Engine implementiert die Empfehlungsalgorithmen, welche bei der Berechnung der Empfehlung verwendet werden. Die Recommender Engine besitzt eine interne Datenbank, welche die Locations samt deren Attribute und die Bewertungen der User speichert. Die Recommender Engine kommuniziert über ein generisches Interface mit der Localina Application.

Die **Localina Application** sendet der Recommender Engine die Bewertungen der Benutzer sowie die Location-Daten und fordert von der Recommender Engine die Prediction an.

Zudem stellt sie eine SOAP Schnittstelle für Client- oder Web-basierte Localina Anwendungen zur Verfügung, welche von den **Localina Frontend Applikationen** wie der Localina iPhone App verwendet werden.

2 Design Contraints

2.1 Technologie

Im Herbst 2009 wurde im Rahmen einer Bachelorarbeit bereits ein erster Prototyp der Localina iPhone App entwickelt. In dieser Bachelorarbeit soll der Prototyp bezüglich Usability und Maintainability optimiert werden. Das iPhone soll daher als technologische Plattform beibehalten werden. Die Applikation muss auf dem aktuellen iPhone Betriebssystem lauffähig sein (ab Version 3).

2.2 Systemarchitektur

Die Localina iPhone App soll ein Showcase für das Empfehlungssystem Localina sein. Die Systemarchitektur und die externen Schnittstellen werden daher durch die bestehende Architektur des Localina Recommender Systems definiert. Architektur und Schnittstellen werden im Kapitel 3 (Umgebung) detailliert beschrieben.

3 Umgebung

3.1 Übersicht

Die *Abbildung I-4* stellt die Systemarchitektur des Empfehlungssystems Localina dar. Der Localina Semantic Recommender und der Duine-basierte Recommender stellen zwei unterschiedliche Implementationen der Localina Recommender Engine dar. Zur Zeit kann die Localina Application ihre Empfehlungen wahlweise von einer dieser beiden Recommender Engines beziehen. Der Duine-basierte Recommender wurde aus zwei Gründen eingerichtet: Zum einen wurde ein Testsystem für die Entwicklung der Localina Plattform benötigt, weil der Localina Recommender immer noch Gegenstand des laufenden Forschungsprojektes ist und zum anderen, um die Ergebnisse des Localina Semantic Recommenders validieren zu können.

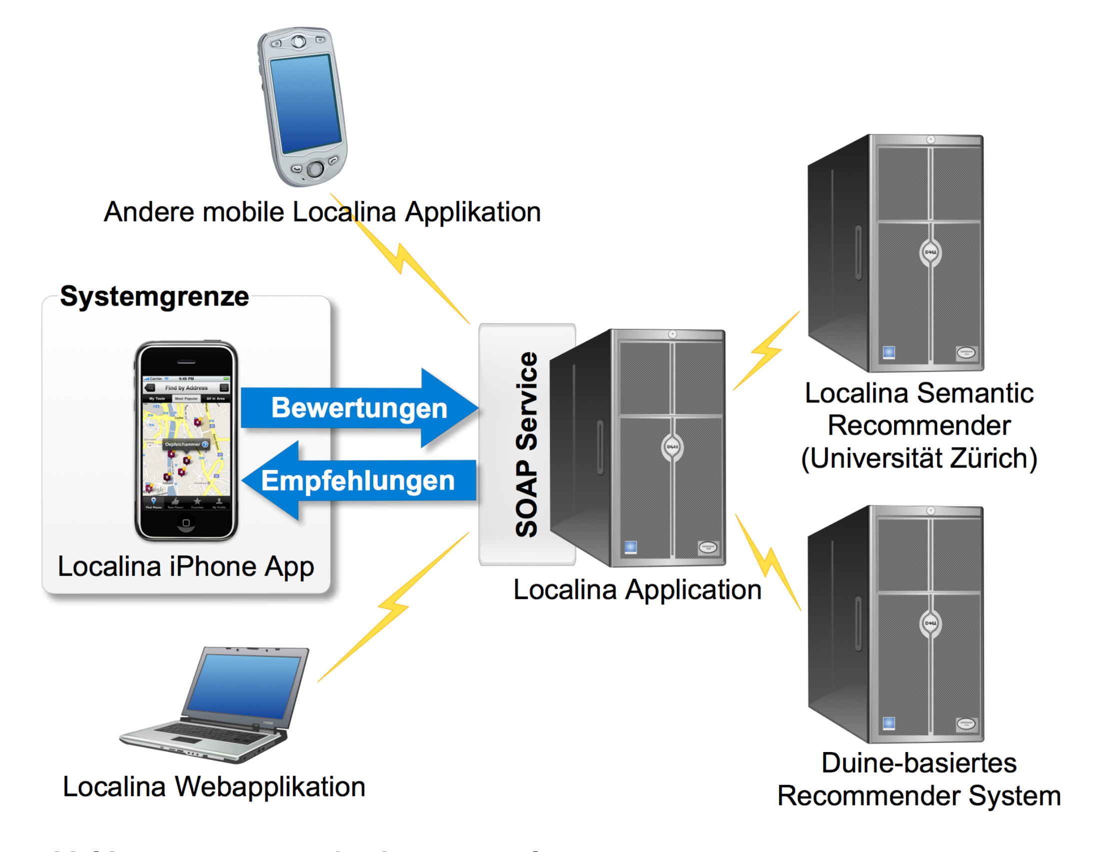


Abbildung I-4: Systemarchitekur von Localina

3.2 Die SOAP-Schnittstelle

In diesem Kapitel wird der SOAP-Service beschrieben, welcher von der Localina Application zur Verfügung gestellt wird. Eine detaillierte Spezifikation kann in Form einer WSDL unter folgender Adresse gefunden werden: http://loap.astina.ch/recommender?wsdl.

3.2.a SOAP-Objekte

Location-Objekt

Das *Location-Objekt* repräsentiert eine Lokalität. Die Attribute des Location-Objekts sind selbsterklärend.

Location-Array

Liste von Location-Objekten.

Prediction-Objekt

Das *Prediction-Objekt* repräsentiert die Prediction zu einer Location. Die folgende Tabelle erklärt die wichtigsten Attribute des Prediction-Objekts.

Attribut	Erklärung	Wertebereich
prediction	Wert der Prediction (Vorhersage des Systems, wie der Benutzer die Location bewerten würde, falls er sie kennt).	Double (-1 < x <= 1)
certainty	Wert für die Verlässlichkeit der Prediction.	Double (-1 < x <= 1)
explanation	Erklärung, wie die Prediction zustande kam.	String
location	Location, zu welcher die Recommender Engine die Prediction errechnet hat.	Location-Objekt

Prediction-Array

Liste von Prediction-Objekten.

Rating-Objekt

Eine Bewertung, welche der User für eine Location abgegeben hat. Die folgende Tabelle erklärt die wichtigsten Attribute des Rating-Objekts.

Attribut	Erklärung	Wertebereich
certainty	Wie sicher ist sich der User in seiner Bewertung.	Double (-1 < x <= 1)
locationId	ID der Location, für welche der Benutzer die Bewertung abgibt.	Long
userIdentifier	Eindeutige Benutzer-ID.	String
value	Wert der Bewertung auf einer Skala von -1 bis 1, wobei 1 die beste Bewertung darstellt.	Double (-1 < x <= 1)

Criteria-Objekt

Kriterien, welche auf eine Location zutreffen müssen, damit sie als Resultat in Frage kommen.

Attribut	Erklärung	Wertebereich
critiquingResults	Falls TRUE, werden nur Locations geliefert, deren Bewertung einen hohen Grenznutzen bringt.	Boolean
distance	Maximale Entfernung in Kilometer vom Standort der Suche.	Double
latitude	Längengrad des Standorts der Suche.	0.0 - ±90.0
longitude	Breitengrad des Standorts der Suche.	0.0 - ±180.0
maxResults	Maximale Anzahl Locations, welche zurückgeliefert werden sollen.	Ganzzahl
searchString	Begriffe, welche in der Beschreibung der Location vorkommen müssen.	String
showAlreadyRatedLocations	Falls FALSE, werden nur jene Locations zurückgeliefert, welche vom User noch nicht geratet wurden.	Boolean
userIdentifier	Eindeutige Benutzer-ID.	String

3.2.b SOAP-Operationen

Die folgende Tabelle gibt einen Überblick über die SOAP-Operationen der Localina Application, welche von der Localina iPhone App verwendet werden.

SOAP-Operation	Erklärung	Request-Format	Response-Format
getLocationsForCriteria	Liefert alle Locations zurück, welche auf das Kriterium (Suchstring) passen.	Criteria-Objekt	Location-Array
getRatingCountForUserIdentifier	Liefert die Anzahl der Bewertungen des Users.	userIdentifier	Ganzzahl
predict	Liefert die Predictions in der gewählten Umgebung zurück.	Criteria-Objekt	Prediction-Array
predictSingleLocation	Liefert die Prediction für eine bestimmte Location zurück.	locationId	Prediction-Objekt
recommend	Übermitteln eines Ratings an die Localina Application.	Rating-Objekt	-
predictLocationsForLocationIdList	Liefert die Prediction für alle Locations der LocationIdList.	String mit kommaseparierten LocationIds	Prediction-Array

4 Stand der Technik

4.1 Konkurrenzvergleich: Restaurant-Guides für das iPhone

4.1.a Tellmewhere

Tellmewhere wurde 2007 von einem französischen Startup gegründet und ist ein Guide für unterschiedlichste Angebote, von Restaurants über Wellnessangebote bis hin zu Touristenattraktionen. Die Empfehlungen von Tellmewhere basieren auf dem sozialen Netzwerk des Benutzers und ein spezieller Matching-Algorithmus erlaubt es ihm, den persönlichen Geschmack mit jenem seiner Freunde zu vergleichen. Der Service bindet daher bestehende soziale Netzwerke wie Twitter und Facebook ein.

Von Tellmewhere gibt es zusätzlich zur Online-Plattform bereits eine ausgereifte Applikation für iPhone und Android. Die Applikation erlaubt es, sich mit Freunden zu vernetzen, nach Restaurants und anderen Freizeitangeboten in der Umgebung zu suchen und Freizeitangebote zu bewerten. Tellmewhere kann wie Localina persönliche Empfehlungen abgeben, indem es die Bewertungen des Users mit denjenigen anderer User vergleicht. Die Restaurantdaten bezieht Tellmewhere von Google. In Zürich scheint es bereits eine ziemlich aktive Community zu geben. Aufgrund der automatisch generierten Benutzernamen (iphone + Laufnummer) lässt sich erkennen, dass die meisten Benutzer von Tellmewhere die iPhone Version der Applikation benutzen.

Vergleich zu Localina

Die Empfehlungen von Tellmewhere basieren ausschliesslich auf Empfehlungen anderer Benutzer und dem Vergleich der Benutzerprofile. Das funktioniert gut, sofern die Benutzerbasis genügend gross ist. In einer kleinen Stadt, wo es nur wenige Benutzer hat, kann jedoch das **Cold-Start Problem** auftreten. Localina kann im Gegensatz zu

Tellmewhere auch Content-basierte Empfehlungen liefern. Das ist ebenfalls ein Vorteil, wenn man sich in einer fremden Stadt aufhält. In diesem Fall gibt es nämlich wenige Überschneidungen mit Präferenzen anderer Benutzer.

Während Tellmewhere Empfehlungen für alle möglichen Konsumprodukte und Freizeitaktivitäten liefert, wird die Localina iPhone App vorderhand nur Empfehlungen für Restaurants liefern. Das könnte jedoch auch durchaus ein Vorteil für Localina sein: Zum einen dürfte es dadurch einfacher sein, das Produkt am Markt zu positionieren und zum anderen kann das Benutzerinterface einfacher gehalten werden, weil es weniger heterogene Daten anzeigen muss.

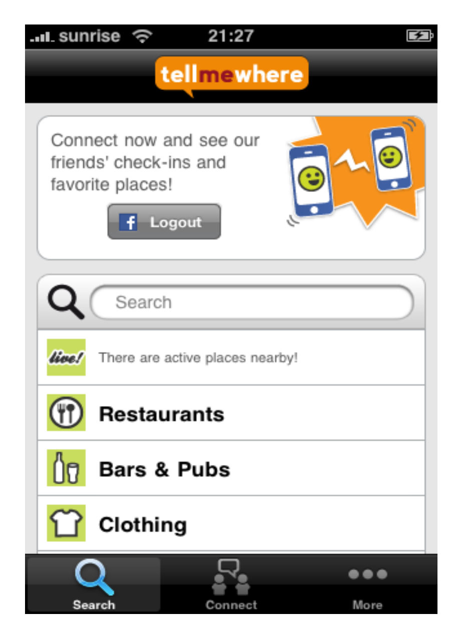


Abbildung I-5: Suche von Tellmewhere

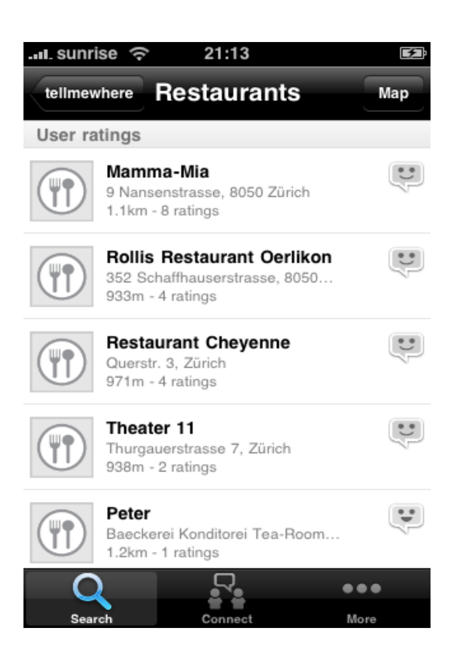


Abbildung I-6: Anzeige der Suchresultate bei Tellmewhere

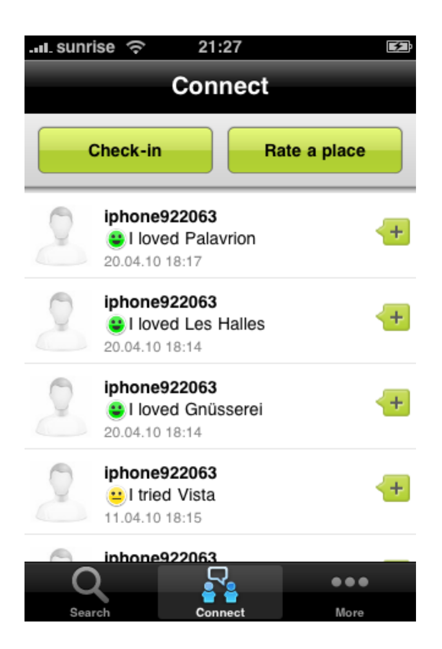


Abbildung I-7: Social Network von Tellmewhere

4.1.b züritipp

Die Redaktion für Zürcher Ausgehtipps des Tages Anzeigers "züritipp", hat im April eine züritipp-App für das iPhone herausgegeben. Die App liefert Informationen zu Veranstaltungen und Restaurants in Zürich und Umgebung.

Vergleich zu Localina

Im Unterschied zu Localina bietet die züritipp-App nebst einem Restaurantführer auch einen weitgefächerten Veranstaltungskalender für die Region Zürich mit den Kategorien Tagestipps, Kino, Musik, Party, Kunst und Bühne. Daneben erhält man auf den Detailseiten zusätzliche Informationen zu Veranstaltungen, ausführliche Besprechungen, Kritiken, Vorankündigungen und Reviews. Der Benutzer der züritipp-App hat die Möglichkeit, sich Restaurants und Veranstaltungen in unmittelbarer Umgebung oder Events nach Datum (heute oder morgen) anzeigen zu lassen. Ebenso können die liebsten Restaurants, Clubs, Museen etc. in eine Favoritenlisten aufgenommen werden.

Localina beschränkt sich bis anhin auf Restaurants. Die Einbindung anderer Arten von Lokalitäten wie Bars, Clubs oder Hotels ist jedoch bereits für einen zukünftigen Ausbauschritt vorgesehen (vgl. Kapitel 6.4 Performance-Tests). Wie züritipp verfügt auch Localina über eine Favoritenliste, die es dem Benutzer ermöglicht, Lieblingslokale zu speichern und bei Bedarf erneut aufzurufen.

Was Localina hingegen grundsätzlich von der züritipp-App unterscheidet, ist das Empfehlungssystem. Nebst der Anzeige von Restaurants in der Umgebung oder an einem bestimmten Ort - wie dies auch bei Züritipp möglich ist -, bietet Localina dem Benutzer den grossen Vorteil, dass er sich auf seinen persönlichen Geschmack zugeschnittene Restaurants anzeigen lassen kann.

Weiter unterscheidet sich Localina von der züritipp-App darin, dass sich die Suche von Lokalen nicht auf die Region Zürich beschränkt. Localina kann Restaurants an einem beliebigen Ort anzeigen.

4.2 Stand der Localina iPhone App

Im Herbst 2009 wurde im Rahmen einer Bachelorarbeit ein erster Prototyp einer nativen iPhone Applikation für Localina entwickelt. Dieser Prototyp nutzt die SOAP-Schnittstelle der Localina Application und hat folgende Funktionen:

- Empfohlene Restaurants im Umkreis der aktuellen Position oder eines gewählten Ortes anzeigen.
- Restaurants bewerten (like, dislike).
- Restaurants auf Karte anzeigen.
- Details zu Restaurant anzeigen.

Als Showcase hat sich der Prototyp bewährt und Astina konnte die iPhone App bereits erfolgreich nutzen, um einen grosser Schweizer Plattformbetreiber vom Localina Recommender überzeugen. Ein Usability-Test hat aber einige Schwachstellen der Applikation aufgezeigt. Der Aufbau und die Ergebnisse dieses Usability-Tests werden im folgenden Abschnitt beschrieben.

4.2.a Usability-Test

Um die Benutzerfreundlichkeit der aktuellen Applikation zu testen, wurde auf Basis des aktuellen Prototypen der Localina iPhone App ein Usability-Test mit potentiellen Benutzern durchgeführt.

Die Versuchspersonen wurden nach folgenden Kriterien ausgewählt:

- Besitzt ein iPhone.
- Geht oft auswärts Essen.
- Wohnt in der Stadt.
- Nutzt das iPhone als Informationsquelle.

Versuchsanordnung

Die Versuche wurden in einem Sitzungsraum bei der Astina AG durchgeführt. Während des Tests wurden Ton- und Videoaufnahmen gemacht. Eine Videokamera wurde auf das Gesicht der Testperson gerichtet, um deren Mimik einzufangen. Eine Dokumentenkamera wurde verwendet, um einen Screencast zu erstellen. Spezielle Beobachtungen wie auffällige Reaktionen und Verhaltensmuster wurden zusätzlich im Testprotokoll notiert.

Vorgehensweise

Der Testperson wurde zuerst die Versuchsanordnung erklärt und die Person wurde darauf hingewiesen, dass Video- und Tonaufnahmen lediglich der späteren Auswertung dienen. Danach wurde ihr der Ablauf des Interviews kurz erläutert und sie wurde zum Mitsprechen aufgefordert (vgl. **Think Aloud**). Die Testperson musste sich die folgende Ausgangssituation vorstellen:

"Eine Freundin hat dir von Localina erzählt, eine Applikation die dir Empfehlungen für Restaurants in Zürich aufgrund deines persönlichen Geschmacks geben kann. Du hast diese Applikation vom iTunes Store heruntergeladen."

Test-Dokumentation

Die Anwendungssituationen, aus welchen die meisten Informationen gewonnen werden konnten, sind in einem Dokument detailliert beschrieben. Diese Dokumente finden sich im Verzeichnis *Usability Tests* auf der Abgabe-CD. Nachfolgend findet sich eine Zusammenstellung der zentralen Aussagen und Beobachtungen.

4.2.b Test mit User 1

Benutzerprofil

- Betriebsökonomin UZH, ledig (Single), 28 Jahre, aus Zürich.
- Geschäftsführerin eines Start-Ups, Label für Herren-Massanzüge und Hemden.
- Kein besonderes Interesse für Technik.
- Geht sehr gerne und oft auswärts essen und kennt daher viele Restaurants.
- Besitzt seit einem Jahr ein iPhone.

Einführung

"Eine Freundin hat dir von Localina erzählt, eine Applikation die dir Empfehlungen für Restaurants in Zürich aufgrund deines persönlichen Geschmacks geben kann. Du hast diese Applikation vom iTunes Store heruntergeladen."

Situationen

"Du gehts an deinem Arbeitsplatz immer etwa in die gleichen Restaurants und du würdest gerne etwas Neues kennenlernen. Stell dir vor du arbeitest hier bei der Astina an der Hardturmstrasse und du möchtest von Localina eine Empfehlung für die nähere Umgebung der Hardturmstrasse."

"Stell dir vor, du hast heute Abend ein Date und du bist für die Auswahl und Reservation des Restaurants verantwortlich, weil er keine Zeit hat. Ihr habt abgemacht, dass ihr euch beim Bellevue trifft. Wie würdest du vorgehen?"

Zentrale Aussagen und Beobachtungen

[Startet Applikation. Nachdem Empfehlungen geladen sind, bewegt sie die Map um sich zu orientieren.] "Wo bin ich jetzt hier? Bin ich jetzt hier wo dieser Mensch eingezeichnet ist?"

"Da sieht man wie weit man schon ist. Das müsste ich jetzt so lange machen, bis es vollständig ist." "Das geht ja gar nie vorwärts."

Interviewer: "Also für ein Date würdest du in ein gleiches Restaurant gehen wie für einen Business Lunch?"

Testperson: "Ja warum nicht. Oder könnte ich das irgendwo ändern?"

4.2.c Test mit User 2

Benutzerprofil

- Informatiker, ledig (Single), 20 Jahre, aus Zürich.
- Geht oft mit Freunden oder der Freundin auswärts essen.
- Besitzt seit zwei Jahren ein iPhone.

Einführung

"Ein Freund hat dir von Localina erzählt, einer Applikation, die dir Empfehlungen für Restaurants in Zürich aufgrund deines persönlichen Geschmacks geben kann. Du hast diese Applikation vom iTunes Store heruntergeladen."

Situationen

"Du gehst mit Freunden in den Ausgang und ihr möchtet gerne zuvor noch was essen. Ihr seid im Niederdorf unterwegs und sucht daher ein Restaurant, das im Niederdorf liegt. Du hast dein iPhone dabei und möchtest Localina benutzen, um ein passendes Restaurant zu finden."

Zentrale Aussagen und Beobachtungen

"Ich finde es gut, dass gleich so eine Beschreibung aufgeht, wo man sehen kann, was die Localina Applikation macht."

Interviewer: "Das Business im Titel irritiert dich nicht?"

Testperson: "Das ist mir jetzt gar nicht aufgefallen." [Überlegt kurz.] "In dem Fall muss ich jetzt zuerst noch meinen Mood einstellen."

Interviewer: "Wie lange würdest du jetzt diese Ratings machen?"

Testperson: “Ich denke, es ist nicht so vorteilhaft, dass man nicht sieht, wie viele Restaurants in der Liste sind. Wenn ich das sehen würde, würde ich denken 'OK, die 10 mach ich schnell'.

Interviewer: “Wie interpretierst du die Anzeige unterhalb der Rating-Boxen?”

Testperson: “Ehrlich gesagt, ist mir die jetzt gar nicht aufgefallen.”

“Mir ist jetzt aufgefallen, dass immer die Restaurants geladen werden, welche sich im Kartenausschnitt befinden, was mir ziemlich gut gefällt.”

4.2.d Test mit User 3

Benutzerprofil

- Männlich, Primarlehrer, verheiratet, 32 Jahre, aus Zürich.
- Technisch nicht so versiert, aber informiert sich gerne über das iPhone.
- Geht über Mittag meist in ein Restaurant und geht regelmässig mit der Frau auswärts essen.
- Besitzt seit einem Jahr ein iPhone.

Einführung

“Ein Freund hat dir von Localina erzählt, einer Applikation, die dir Empfehlungen für Restaurants in Zürich aufgrund deines persönlichen Geschmacks geben kann. Du hast diese Applikation vom iTunes Store heruntergeladen.”

Situationen

“Du möchtest heute Abend mit deiner Frau in ein Restaurant in Zürich gehen. Du suchst nach etwas romantischem. Weil dir gerade kein gutes Restaurant einfällt, möchtest du schauen, was dir Localina empfiehlt.”

Zentrale Aussagen und Beobachtungen

[Klick auf einen Pin und sieht sich die Details zum Restaurant an.] "Da sehe ich die Informationen. Leider gibt es nicht sehr viele Informationen."

[Wechselt von der Detailansicht zurück zur Kartenansicht.] "Jetzt lädt es schon wieder die persönlichen Empfehlungen."

Interviewer: "Eigentlich sollte man nur jene Restaurants bewerten, welche man kennt."

Testperson: "Da gibt es ja keinen 'Cancel-Button'."

"Ich habe jetzt ein paar indische und ein paar asiatische bewertet und erwarte, dass ich ähnliche Vorschläge kriege."

5 Bewertung

5.1 Konkurrenzvergleich

In der folgenden Tabelle werden die Funktionen der Localina iPhone App mit den Konkurrenz-Applikationen von Tellmewhere und züritipp verglichen.

	Tellmewhere	züritipp	Localina
Personalisiert Empfehlungen mittels Collaborative Filtering.	✓		✓
Personalisierte Empfehlungen aufgrund von Präferenzen zu bestimmten Arten von Lokalitäten.			✓
Restaurants in der Umgebung finden.	✓	✓	✓
Restaurants an einem Ort nach Wahl finden.		✓	✓
Empfehlungen der Freunde ansehen.	✓		✓
Lokalitäten zu Favoriten hinzufügen.		✓	✓
Detaillierte Beschreibung der Lokalitäten.		✓	

5.2 Stand des iPhone Client Prototyps

Aus den Resultaten des ersten Usability Tests lassen sich folgende Resultate ableiten:

Positiv	Negativ
• Die Bedienung des Rating Interface ist den meisten sofort klar. • Es ist klar, welche Funktion hinter welchem Tab steckt. • Wie man Moods editiert und auswählt ist intuitiv. • Umgang mit der Map (Zoom, Suche, Klick auf Pins) ist vertraut. • Die Information in der Infobox ist verständlich und den Usern ist nach dem Lesen klar, dass sie zuerst Bewertungen abgeben müssen.	• Prozess (Mood auswählen > Raten > Empfehlungen anzeigen) ist nicht klar. • Die Bedeutung der Nummern ist nicht klar. • Die Fortschrittsanzeige ist zu unauffällig und zu wenig aussagekräftig. • Der Zusammenhang zwischen dem Mood und den Ratings ist nicht genügend nachvollziehbar. • Zu Beginn nicht klar, was der Kartenausschnitt anzeigt. Es fehlt eine Erklärung. • Empfehlungen werden zu oft neu geladen, wobei für eine Weile die gesamte Map blockiert wird. • Die Benutzer wollen sehen, wie viele Restaurants sie bewerten sollen. • Die Infobox ist zu wenig aufdringlich und wird nur von den wenigsten Benutzern gelesen.

6 Vision

6.1 Anforderungen

6.1.a Hintergrund des Projekts

Der Wirtschaftspartner, die Firma Astina AG mit Sitz in Zürich, entwickelt zusammen mit der Universität Zürich im Rahmen eines KTI Projektes Localina, ein Empfehlungssystem für Restaurants, Bars, Cafés, Hotels und Clubs. Das Team der Universität Zürich entwickelt dabei den Empfehlungsalgorithmus und Astina ist für die Entwicklung der Plattform verantwortlich. Astina möchte Localina als Showcase nutzen, um potentiellen Kunden eine Anwendungsmöglichkeit für das Empfehlungssystem an einem anschaulichen Beispiel präsentieren zu können. Zur Ergänzung der aktuellen Web-basierten Demonstrationsanwendung wurde im Herbstsemester 2009 ein erster Prototyp eines iPhone Clients für Localina entwickelt.

6.1.b Ziele

Die Localina iPhone App soll das Localina Recommender System nutzen, um persönliche Empfehlungen für Restaurants liefern zu können. Die Kommunikation zwischen der iPhone App und dem Localina Recommender System soll über die bestehende SOAP-Schnittstelle erfolgen, welche von Astina bereitgestellt wird. Damit Localina persönliche Restaurantempfehlungen generieren kann, muss der Benutzer eine genügende Anzahl Restaurants bewerten. Die Applikation muss ihm daher ein Interface bereitstellen, um ihm bekannte Restaurants zu bewerten. Es muss auch klar erkennbar sein, warum diese Bewertungen wichtig sind und was der Impact auf die Qualität der Empfehlungen ist. Diesbezügliche Schwachstellen der ersten Version sollen in der aktuellen Version verbessert werden.

6.1.c Anforderungen seitens Astina

Wie eingangs erwähnt, möchte Astina die Localina iPhone App als Showcase für ihr Empfehlungssystem nutzen. Daher muss klar ersichtlich sein, inwiefern sich die Localina iPhone App von herkömmlichen Bewertungssystemen und Restaurant-Guides unterscheidet und wieso die Empfehlungen von Localina genauer sind als deren, meist auf dem Durchschnittsgeschmack basierenden Ratings. Eventuell möchte Astina die iPhone App später im iTunes App Store publizieren. Die Applikation muss daher damit verbundene Anforderungen an das Design des User-Interfaces und die Codequalität erfüllen.

6.1.d Benutzeranforderungen

- Erkennbarer Mehrnutzen gegenüber herkömmlichen Restaurant Guides.
- Es soll auf den ersten Blick erkennbar sein, welche Restaurants empfohlen sind.
- Inuitiv zu bedienen.
- Wenige Schritte, um zur persönlichen Empfehlung zu gelangen.
- Schneller und nachvollziehbarer Lernfortschritt des Systems.
- Suchen von Empfehlungen an einem bestimmten Ort.
- Nützliche Infos zu den Restaurants.

6.1.e Nichtanforderungen

Im Gegensatz zum Localina Recommender System muss die iPhone App nur Restaurants und keine weiteren Arten von Lokalitäten unterstützen. Idealerweise wird bei dem Redesign jedoch eine spätere Integration von weiteren Kategorien berücksichtigt.

6.2 Vision Statement

Die Applikation soll nicht nur ein reiner Showcase sein, sondern später auch im iTunes App Store publiziert werden können. In einer weiteren Version sollen auch zusätzliche Kategorien von Lokalitäten hinzugefügt werden. Längerfristig wird angestrebt, dass sich Localina als Empfehlungssystem für Lokalitäten etabliert und eine breite Benutzerbasis erlangt.

6.3 Product Backlog

Die folgenden User Stories beschreiben die Anforderungen an die Localina iPhone App. Die User Stories werden gemäss ihrem Business Value (Geschäftswert) priorisiert. Priorität A steht für "must have", B für "can have" und C für "nice to have".

ID	User Story	Beschreibung	Business Value
US01	**Applikation starten**	Bei Starten der Applikation erscheint ein Splash Screen.	C
US02	**Als Benutzer möchte ich auf den ersten Blick sehen, welche Funktionen mir die Applikation bietet.**	Nach dem Start erscheint eine Übersichtsseite, welche die nächsten Schritte aufzeigt.	B
US03	**Als Benutzer möchte ich die Lokalitäten auf der Karte betrachten können.**	Die Empfehlungen werden direkt auf einer Karte angezeigt.	A
US04	**Als Benutzer möchte ich die Lokalitäten in einer Liste betrachten können.**	Wenn man sich auf der Karte mit den Empfehlungen befindet, kann man die Karte umdrehen und findet dort die angezeigten Empfehlungen in einer Liste.	B
US05	**Als Benutzer möchte ich eine Lokalität in der Nähe finden.**	Lokalisierung des aktuellen Standorts mittels LBS.	A

ID	User Story	Beschreibung	Business Value
US06	**Als Benutzer möchte ich eine Lokalität in einer bestimmten Stadt finden.**	Optionen: Stadt als Suchbegriff eingeben (Search-As-You-Type liefert Vorschläge aus bereits gesuchten Begriffen). Stadt aus einer Liste der 5 zuletzt gesuchten Städte auswählen.	C
US07	**Als Benutzer möchte ich eine Lokalität im Umkreis eines anderen Standorts als meines aktuellen finden.**	Optionen: Adresse als Suchbegriff eingeben (Search-As-You-Type liefert Vorschläge aus bereits gesuchten Begriffen). Adress aus einer Liste der 5 zuletzt gesuchten Adressen auswählen.	A
US08-1	**Als Benutzer möchte ich sehen, welche Lokalitäten meinem persönlichen Geschmack entsprechen.**	Empfohlene Restaurants werden mit einem speziellen Symbol gekennzeichnet.	A
US08-2	**Als Benutzer möchte ich sehen, welche Lokalitäten meinem persönlichen Geschmack entsprechen.**	Der Benutzer kann einen Filter setzen, womit nur jene Restaurants angezeigt werden, die seinem persönlichen Geschmack entsprechen.	B
US09	**Als Benutzer möchte ich sehen, wie beliebt eine Lokalität bei den anderen Benutzern ist.**	Bei jeder Lokalität wird deren durchschnittliche Bewertung angezeigt.	B
US10	**Als Benutzer möchte ich Informationen zu einer Lokalität ansehen können.**	Zu jeder Lokalität kann eine Detailansicht angezeigt werden.	A
US11	**Als Benutzer möchte ich bei einem Restaurant direkt einen Tisch reservieren können.**	Integration des Tischreservationssystems OpenTable.com.	C
US12	**Als Benutzer möchte ich die Speisekarte der Restaurants einsehen können.**	Falls die Lokalität eine Online-Speisekarte hat, kann man von der Detailansicht direkt auf eine Ansicht wechseln, wo diese dargestellt wird.	C

US13	**Die Benutzer sollen regelmässig daran erinnert werden, Lokalitäten zu bewerten.**	Es gibt eine Rating-Inbox, wo dem Benutzer regelmässig Lokalitäten zum Bewerten vorgeschlagen werden. Die Anzahl Items in der Rating-Inbox werden sowohl auf dem Rating-Tab sowie auf dem Applikations-Icon auf dem iPhone Desktop angezeigt. Es werden Lokalitäten vorgeschlagen, welche der Benutzer mit hoher Wahrscheinlichkeit kennt.	B
US14	**Als Benutzer möchte ich eine Lokalität direkt in der Detailansicht bewerten können.**	Rating-Interface in der Detailansicht einer Lokalität.	B
US15	**Als Benutzer möchte ich sehen, wieviele Lokalitäten ich bewerten muss, um gute Ergebnisse zu erhalten.**	Eine Anzeige visualisiert dem Benutzer, wie gut der Personalisierungslevel von Localina ist und gibt ihm einen Hinweis, wie er diesen verbessern kann.	B
US16	**Als Benutzer möchte ich mir den Anfahrtsweg zu einer Lokalität anzeigen können.**	Aufruf des entsprechenden Dienstes in Google Maps.	C
US17	**Fehlermeldung, falls der Localina-Server nicht erreichbar ist.**	Alert anzeigen.	C
US18	**Fehlermeldung anzeigen, falls keine Internetverbindung verfügbar ist.**	Alert anzeigen.	C
US19	**Ladeanzeige, wenn Daten über das Internet geladen werden.**	Aktivitätsanzeige in der Statusliste. Mit Vorteil sollte die Ladeanzeige die Benutzeroberfläche nicht blockieren.	A
US20	**Als Benutzer möchte ich ein einzelnes Restaurant auf der Karte anzeigen können.**	Beim Klick auf die kleine Map in der Detailansicht öffnet sich eine Grossansicht der Map, welche die Lokalität darstellt.	C
US21	**Falls mehrere Standorte auf eine Adresssuche zutreffen, soll eine Auswahlliste angezeigt werden.**	Beispielsweise als Popup (wie bei Google Maps).	C
US22	**Die Adressen in der Search History sollen formatiert dargestellt werden, so dass ähnliche Eingaben aufeinander gemappt werden können.**	Bei einer Anfrage an den Google Geocoder-Service kommt eine formatierte Addresse zurück, welche gespeichert wird.	C

US23	**Custering der Annotations bei geringem Zoomlevel.**	Wenn es in einer Umgebung sehr viele Lokalitäten hat, wir die Darstellung schnell unübersichtlich. Map-Pins, welche nahe zusammenliegen, sollten zusammengefasst und mit einem speziellen Icon als Cluster gekennzeichnet werden. Bei einem Klick auf das Cluster, soll die Map so nahe heranzoomen, dass die einzelnen Lokalitäten sichtbar sind.	C
US24	**Paging für Resultatliste (analog Clustering in Map).**	Wenn es in einem Kartenausschnitt zu viele Resultate gibt, soll ein Paging möglich sein.	C

6.4 Anwendungsumfeld

Das Anwendungsumfeld wird nachfolgend anhand von Personas und Szenarios analysiert.

6.4.a Personas

Anna, die Social Media Managerin

„Mich stört der Durschnittszürcher, ich habe es lieber einzigartig anders."

Die 37-jährige Anna arbeitet als Social Media Managerin bei einer Kommunikationsagentur in Zürich. Anna lebt zusammen mit ihrem Lebenspartner Kurt, der als Vermögensberater bei einer kleinen Privatbank arbeitet. Das kinderlose Paar wohnt in einer stilvoll eingerichteten Attikawohnung im Zürcher Kreis 5. Anna ist beruflich sehr ehrgeizig und arbeitet oft bis spät am Abend. Wenn es ihre spärliche Freizeit erlaubt, geht sie am Abend gerne ihrem Hobby, dem Kochen nach. Noch lieber entdeckt sie zusammen mit Kurt neue Restaurants. Bei der Wahl des Restaurants folgt Anna meistens den Empfehlungen ihrer Freundinnen oder wagt es einfach mal spontan ein ihr unbekanntes, kleines Restaurant zu testen. Da Anna selbst gerne kocht und auf dem Land aufgewachsen ist, möchte sie gesundes, saisonales Gemüse auf dem Teller haben. Überhaupt lebt Anna sehr gesund und trainiert zwei mal in der Woche im Fitnesscenter oder geht früh morgens vor der Arbeit eine Stunde joggen.

Anna besitzt zwei Mobiltelefone. Ihr Arbeitgeber stellt ihr für die geschäftliche Kommunikation einen BlackBerry zur Verfügung. Privat nutzt sie jedoch lieber ihr iPhone. Anna ist sehr erfahren im Umgang mit dem Internet und mit mobilen Kommunikationsgeräten, denn die gehören zu ihrem beruflichen Alltag. Am iPhone schätzt sie die besonders das Design und die Einfachheit, denn sie benutzt es häufig während Aktivitäten wie Autofahren, Einkaufen oder Joggen.

Maria, die Mutter

„Ich liebe den Sommer in Zürich mit dem Kind im Park"

Maria ist 30 Jahre alt und arbeitete bis vor kurzem als Assistentin bei einer Zürcher Anwaltskanzlei. Maria ist alleinerziehende Mutter einer Tochter und wohnt in einer Wohnung in Winterthur. Maria lernt gerne Leute kennen und vermisst die wilden Partynächte von früher, als sie noch nicht die Verantwortung trug eine Tochter gross zu ziehen. Sie würde gerne Teilzeit arbeiten, findet jedoch keinen Krippenplatz. Die Krippenplatz-Suche ist anstrengend und diese Umstände verlangen von Maria, dass sie ihre Zeit und vor allem ihr Budget genau plant.

Maria trifft sich oft mit ihren Freundinnen, ebenfalls junge Mütter, in Zürcher Grünanlagen. Oft besprechen sie neue, praktische iPhone Apps und die Geheimtipps der Zürcher Gastroszene.

Johny, der Student

„Nichts schöneres als mit Freunden über Technik zu plaudern"

Johny (23) hat gerade mit seinem Elektrotechnik-Studium an der ETH Zürich angefangen. Er wohnt in Zürich Oerlikon, ist Vegetarier und meint, dass der Konsum von Fleisch ökologisch nicht zu verantworten ist. Zudem ist er der Meinung: "Frauen stehen auf Vegetarier". In seiner Freizeit macht er gerne Sport, geht ins Kino oder trifft sich mit Freunden in einer Bar um über neue Gadgets und Technik zu plaudern. Als Single hat er öfter ein Date. Als Gentleman sieht er es natürlich als seine Pflicht, die Wahl des Lokals und die Tischreservation selbst zu organisieren. Er kennt bereits einige Geheimtipps von romantischen Restaurants, wo er regelmässig hingeht. Es muss ruhig, gemütlich und darf auch nicht zu teuer sein. Tagsüber geht er lieber in eine Mensa und spart gerne beim Essen um dann am Abend in seine Dates zu investieren.

Für Jonny muss alles funktional sein und er ist der Ansicht, dass das iPhone das stabilste und fortschrittlichste Smartphone ist, welches nie abstürzt. Durch sein Studium hat er grosses Interesse an technischen Geräten und ist sehr geübt Umgang mit diesen.

6.4.b Szenarios

Szenario Anna

Es ist 10 Uhr morgens. Anna trifft sich am Mittag mit ihrem Chef zum Lunch, um ein internes Projekt zu besprechen. Meistens gehen Sie über Mittag mit ihren Arbeitskollegen/innen ins Lily's, weil es nah und gut ist. Gestern hat ihr eine Arbeitskollegin jedoch von der iPhone Applikation "Localina" erzählt. Dort erhalte man Restaurant Empfehlungen, die auf den indivuellen Geschmack des Benutzers zugeschnitten sind. Anna hat sich die Applikation bereits vom iTunes Store heruntergeladen und möchte nun gerne sehen, ob ihr Localina tatsächlich was gutes empfehlen kann. Obwohl sie eigentlich nicht viel Zeit hat, startet sie die Applikation und sieht sich Empfehlungen in ihrer Nähe an. Nun sieht sie die Anzahl der Locations aufleuchten die in ihrer Nähe sind und sie noch zu bewerten hat. Ihrem Ordnungssinn entsprechend macht sie sich sofort daran diese Restaurants kurz anzusehen und diejenigen die sie kennt zu bewerten. Nach wenigen Bewertungen lässt Anna sich dann die Empfehlungen anzeigen, hat aber nicht die Zeit alle durchzugehen. Die ersten 3 empfohlenen Restaurants kennt sie bereits - darunter auch das Lilly's welches sie bereits bewertet hat und sie erkennt, dass das Restaurant nun das Icon mit ihrer Bewertung bekommen hat. Sie entscheidet sich für das Restaurant Shaba, da sie weiss, dass ihr Chef asiatisch (wie auch das Lily's) mag und eine Freundin mal erwähnt hat, dass das Restaurant ein sehr stilvolles Ambiente habe.

Szenario Maria

Maria ist mit ihren Freundinnen und den Kindern im Park und sie überlegen sich, wo sie heute Abend mit den Kindern etwas gesundes und günstiges essen könnten. Es ist Freitag und normalerweise sind die Restaurants zu gut besetzt um eine lärmende Schaar Kindern wohlwollend zu begrüssen. Deshalb würden sie sich gerne etwas von einem Take-Away holen und im Park gemütlich und unkompliziert geniessen. Die Suche gestaltet sich nicht einfach, bis eine Freundin von der Localina Anwendung erzählt und die jungen Mütter diese ausprobieren. Sie bewerten die Take-Aways die sie schon kennen und suchen darauf

auf der Google Maps Karte Grünflächen in Zürich um zu sehen, was in der Nähe empfohlen wird.

Szenario Jonny

An einer WG-Party, welche Jonny und sein Mitbewohner organisiert haben, hat sich Jonny länger mit einer Freundin seines Mitbewohners unterhalten. Die beiden haben viel über Sport und Technik gesprochen und wollen morgen Abend zusammen was essen und dann ins Kino in der Nähe des Bellevue gehen. So startet Jonny seine Localina Applikation und tippt “bellevue zurich” in die Suchbox unter "Find Locations" ein. Er benutzt Localina nicht zum ersten Mal und hat daher bereits einige Dutzend Bewertungen von romantischen Restaurants gemacht. Es beruhigt ihn, dass ihm sowohl das Rosaly's wie auch das Cakefriends von Localina empfohlen werden. Sein Vertrauen in die Empfehlungen ist daher gross. Er sieht, dass das Odéon an zweiter Stelle liegt. Er geht in die Detailansicht und wählt die Telefonnummer, um einen Tisch für zwei zu reservieren.

II Projektdokumentation

1 Vision

Der Inhalt des Vision Dokument ist im *Kapitel I-6 Vision* zu finden.

2 Anforderungsspezifikation

2.1 Analyse der Ausgangslage

In einer ersten Analysephase wurde der Stand des aktuellen Prototypen und die Interfaces zur Localina Server Anwendung dokumentiert. Die Resultate sind in den Kapitel Die SOAP-Schnittstelle und Stand der Localina iPhone App zu finden.

2.2 Zielgruppe

Die Localina iPhone Applikation richtet sich primär an Einwohner von Zürich, welche gerne und oft auswärts Essen gehen und somit viele Restaurants kennen. Dies trifft hauptsächlich auf Personen der Altersgruppe zwischen 25 und 55 Jahren zu, welche über ein genügendes Einkommen verfügen. Weil es später möglich sein soll, die Anwendung auch mit Bars und Clubs zu erweitern, sollte jedoch auch eine jüngere Zielgruppe berücksichtigt werden.

2.3 Redesignentscheide

Aus den Testbeobachtungen wurden wurden folgende Redesignentscheide abgeleitet.

Problem	Lösung (Redesign-Entscheid)
Prozess (Mood auswählen > Raten > Empfehlungen anzeigen) ist nicht klar.	Das Konzept der Moods hat zu mehr Verwirrung geführt, als es einen Nutzen hat. Die Idee der Moods ist daraus entstanden, dass die Leute in unterschiedlichen Situationen unterschiedliche Vorlieben haben. Die Idee war, dass sie ihre Bewertungen innerhalb eines Moods (z.B. Romantic) abgeben und damit Empfehlungen erhalten, welche dem Mood entsprechen. Weil das Konzept der Moods den meisten Usern sowieso nicht verständlich ist und kaum jemand beim Bewerten Rücksicht auf den Mood nehmen wird, soll das Konzept weggelassen werden.
Die Bedeutung der Nummern ist nicht klar. Ist 1 die beste Bewertung oder 10? Sind alle empfohlen oder nur die höchsten Nummern?	Die Nummern sollen durch aussagekräftigere Symbole ersetzt werden.
Die Fortschrittsanzeige ist zu unauffällig und zu wenig aussagekräftig.	Die Bedeutung der Fortschrittsanzeige soll z.B. mittels einer Sprechblase erklärt werden.
Zu Beginn ist nicht klar, was der Kartenausschnitt anzeigt. Es fehlt eine Erklärung.	Die Navigation soll so überarbeitet werden, dass der Benutzer selbst entscheidet, was er als erstes machen will. Wenn er selbst wählt, dass er die Empfehlungen in seinem Umkreis anzeigen möchte, weiss er auch genau, was der Kartenausschnitt anzeigt.
Empfehlungen werden zu oft neu geladen, wobei für eine Weile die gesamte Map blockiert wird.	Der Benutzer soll selbst entscheiden, wann er die Empfehlungen neu laden möchte.
Die Benutzer sehen nicht, wie viele Restaurants sie bewerten sollen.	Die Applikation soll dem Benutzer aktiv Tasks zum Bewerten geben.
In einer typischen Situation, in der man Localina verwenden würde, hat man keine Lust oder Zeit, Restaurants zu bewerten.	Der Benutzer soll in Situationen, wo es ihm langweilig ist darauf aufmerksam gemacht werden, dass er noch Restaurants für Localina bewerten soll.
Die Infobox ist zuwenig aufdringlich und wird von den wenigsten Benutzern gelesen.	Infobox soll beim ersten Start der Applikation angezeigt werden. Zudem soll die Infobox übersichtlich gestaltet sein und nicht zuviel Text enthalten, dass sie von den Benutzern auch gelesen wird.

2.4 Minimale Anforderungen

Da eine Erweiterung des Funktionsumfangs des aktuellen Localina iPhone Clients nicht Teil dieses Projektes ist, leiten sich die minimalen Anforderungen aus den Funktionen des Prototyps ab. Die minimalen Anforderungen werden nachfolgend in Form von Brief Use Cases beschrieben.

2.4.a Restaurant bewerten

ID	FR-01
Beschreibung	Der Benutzer muss ein Restaurant bewerten können.

2.4.b Empfehlungen im Umkreis des aktuellen Standorts

ID	FR-02
Beschreibung	Der Benutzer muss die Empfehlungen im Umkreis des aktuellen Standorts auf einer Karte anzeigen können.

2.4.c Empfehlungen an einer bestimmten Adresse anzeigen

ID	FR-03
Beschreibung	(1) Der Benutzer gibt eine Adresse in ein Suchfeld ein. (2) Die Applikation zeigt die Empfehlungen im Umkreis gewählten Adresse auf der Karte an.

2.4.d Details zu einem Restaurant

ID	FR-04
Beschreibung	Der Benutzer kann Detailinformationen zum Restaurant anzeigen. Diese sollen in einer separaten Detailansicht dargestellt werden.

2.5 Übrige Anforderungen

Während der Iteration 1 (Optimierung des externen Designs) wurde ein Backlog geführt (vgl. Kapitel 6.2 Product Backlog). Die User Stories des Backlog wurden gemäss ihrem Geschäftswert priorisiert. Alle User Stories mit einer Priorität von A oder B sollten bis zum Ende der Iteration 1 implementiert werden.

3 Externes Design

3.1 Einleitung

In diesem Kapitel soll die Entwicklung des externen Designs von Localina von den einzelnen Mockup Versionen bis hin zum ersten Release, sowie die den Designänderungen zugrunde liegenden Überlegungen aufgezeigt und erläutert werden.

3.2 Mockups

Das externe Design wurde anhand von Papierprototypen schrittweise entwickelt. In diesem Abschnitt wird die Entwicklung der Papierprototypen anhand der wichtigsten Änderungen beschrieben. Die vollständigen Papierprototypen der Mockup Versionen 1 bis 2.2 sind im Verzeichnis *Mockups* auf der Abgabe-CD zu finden.

3.2.a Mockup Version 1

Das vollständige Mockup findet sich unter *Mockups/Mockup-v1.pdf* auf der Abgabe-CD.

Vorschläge zum Raten und rote Ziffernanzeige

Dieser Phase entstammt die Idee der oberhalb des Localina-Icons eingeblendeten roten Ziffernanzeige (vgl. Nummer 1 auf Seite 1 des PDF).

Die Ziffer auf dominant-farbenem Hintergrund soll den Benutzer von Zeit zu Zeit zum Bewerten von Restaurants animieren. Startet der User die Applikation wird er durch Wiederfinden derselben Ziffernanzeige auf dem Tab *Rate* direkt zum Bewertungsmodus geführt. Dabei macht sich Localina die Gegebenheit zunutze, dass viele User die Ziffer in der roten Anzeige möglichst tief halten wollen.

Neue Rubriken

Die Rubriken (Tabs) wurden geändert. Neu eingeführt wurde die Rubrik *Search*, welche nachfolgend erläutert wird. Weil das Register *Info* beim aktuellen Prototypen der Localina iPhone App leer war, wurde es weggelassen. Dafür gibt es eine neue Rubrik *Toplist*, welche die persönlichen Empfehlungen in einer Liste anzeigt. Das Icon für die Toplist ist das selbe wie der Map-Pin, welcher die Empfehlungen auf der Karte anzeigt, um den Zusammenhang zu verdeutlichen. Das Register *Your Top 10* mit der Kartenansicht der Empfehlungen wurde umbenannt in *Map*.

Register Search

Das Register *Search* (vgl. Seiten 2 und 3 des PDF) wurde eingeführt, um dem User einen klareren Einstieg in die Applikation zu geben. Beim aktuellen Prototypen landet der User nach dem Start der Applikation direkt in der Map und weiss nicht, warum genau dieser Kartenausschnitt angezeigt wird und was die Map-Pins darstellen.

Unter *Search* kann ein Restaurant nach den Kriterien *Nearby*, *By City* und *By Address* gesucht werden. Startet man eine Suchanfrage für eine bestimmte Stadt oder Adresse, erscheinen unterhalb des Suchfeldes die bereits einmal gesuchten Städte oder Adressen (*Recent cities*, *Recently visited*). Diese können direkt angewählt werden. Dem User soll damit erspart werden, eine bereits einmal ans System gesandte Suchanfrage erneut im Suchfeld eintippen zu müssen.

Neue Map-Pins

Unter *Map* werden dem Benutzer wie bis anhin die Restaurants in seinem Umkreis oder an einem gewählten Ort angezeigt. Bisher wurden dem User nur die 10 besten Restaurants angezeigt, welche alle das gleiche Symbol hatten. Der User konnte somit nicht erkennen, ob alle empfohlen sind oder nur einzelne davon.

Neu gibt es drei verschiedene Kennzeichnungen für die Restaurants (vgl. Seite 4 des PDF). Die Bedeutungen sind wie folgt:

- Das Standard-Icon ist eine blaue Sprechblase mit einem Restaurant-Symbol.

- Alle Empfehlungen sind zusätzlich mit einem Stern versehen.
- Restaurants, zu welchen das System keine Bewertung abgeben kann (zu wenige Attribute oder Bewertungen) werden grau dargestellt.

Bewerten in der Detailansicht

In der Detailansicht (vgl. Seiten 5 und 6 im PDF) gibt es 5 Smilie-Buttons, über welche man direkt eine Bewertung abgeben kann. Falls man das Restaurant bereits kennt, kann man es also direkt mittels Antippen des passenden Smileys bewerten. Wurde das Restaurant bereits bewertet, ist der entsprechende Button vorselektiert.

Informationen zur Empfehlung

Solange das System noch keine verlässlichen Empfehlungen abgeben kann, wird der Benutzer über den Button *Improve Localina!* aufgefordert, zuerst einige Restaurants zu bewerten (vgl. Seite 5 im PDF). Hat man ein empfohlenes oder aber ein nicht empfohlenes Restaurant angetippt, erscheint in der Informationsansicht ein eingefärbtes Smiley, das den Durchschnitt der Bewertungen anderer User anzeigt. Zudem wird der Rang angezeigt, welcher dem User aufzeigen soll, wie gut das angetippte Restaurant im Vergleich zu den anderen 24 erschienenen Restaurants abschneidet.

Register Toplist

Die *Toplist* zeigt dem Benutzer auf einen Blick die seinem Geschmack entsprechenden Restaurants an, sofern er bereits genügend Bewertungen abgegeben hat. Ansonsten erscheint eine Aufforderung zum Bewerten von Restaurants.

Neues Rating-Interface

Das Problem des Rating-Interface des aktuellen Prototypen war, dass der Benutzer keine Übersicht hatte, wieviele Restaurants er bewerten soll und wann er damit aufhören kann. Daher werden die zu bewertenden Restaurants neu in Form einer Task-Liste dargestellt (vgl. Seite 9 des PDF).

Bewertungen können entweder anhand der fünf Smileys oder eines Fragezeichens vorgenommen werden. Mit dem Fragezeichen kann man eine Bewertung auslassen, falls man das Restaurant nicht kennt. Hat man eines der Restaurants bewertet, wird das gewählte Smiley eingefärbt und das Restaurant wird in den Register *Already Rated* verschoben.

3.2.b Mockup Version 2

Die erste Mockup-Version wurde anhand von kleinen Benutzerbefragungen auf Basis des Papierprototypen validiert, wobei sich einige Schwächen aufgezeigt haben. In der Mockup Version 2 wurde versucht, diese Schwächen zu beheben. Das vollständige Mockup findet sich unter *Mockups/Mockup-v2.pdf* auf der Abgabe-CD.

Änderung der Register (Tabs)

Der Zusammenhang zwischen einer Suche (*Register Search*) und den Registern *Map* und *Toplist* war nicht genügend klar. Daher wurden die drei Register zusammengefasst, um zu verdeutlichen, dass die drei konzeptionell zusammengehören und dass sowohl die Map wie auch die Liste die Resultate der Suche anzeigen.

Startscreen

Die Informationen zur Funktionsweise von Localina wurden weggelassen, weil ihnen die Testpersonen keine Beachtung schenkten. Neu werden nur noch drei Buttons angezeigt, womit der Benutzer sich schneller einen Überblick verschaffen kann.

Map-Filter für Your Tastes und Most Popular

Damit der Benutzer deutlicher sieht, welche Restaurants seinem persönlichen Geschmack entsprechen und welche Restaurants bei den anderen Benutzern am beliebtesten sind, wurden Filter eingeführt (vgl. Seite 4 des PDF). Unter *Your Tastes* werden nur die personalisierten Empfehlungen dargestellt und unter *Most Popular* die Restaurants, mit der besten durchschnittlichen Bewertung. Unter *All in Area* werden alle Suchresultate angezeigt.

Neue Map-Pins

Um die Aussagekraft der Icons zu verbessern, wurden in Anlehnung an die Iconographie von Tellmewhere folgende Neuerung eingeführt (vgl. Seiten 8 und 9 im PDF):

- Runde Icons kennzeichnen Restaurants, welche der User selbst bewertet hat (seine eigene Meinung).
- Smilies wiederspiegeln die Meinung von Localina.

Ignore Selected Ratings

Weil das Fragezeichen zum Auslassen einer Bewertung den Testpersonen nicht klar war, wurde das System dahinter geändert. Dieses wurde an die Lösch-Funktion der E-Mail Applikation des iPhone angelehnt, womit die Bedienung dem Benutzer vertraut sein sollte.

Register Info

Statt der Informationen auf der Einstiegsseite wurde eine separate Rubrik für Informationen über den Lernstatus von Localina und wie Localina funktioniert eingeführt.

3.2.c Mockup Version 2.1

Die Ergebnisse der zweiten Mockup-Version wurden anhand von einem Usability-Test validiert. Die Ergebnisse flossen in eine weitere Mockup-Version ein. Das vollständige Mockup findet sich unter *Mockups/Mockup-v2.1.pdf* auf der Abgabe-CD.

Umbenennung der Register (Tabs)

Die Register *Search* und *Rate* wurden in *Find Locations* respektive *Rate Locations* umbenannt, damit deren Bedeutung klarer wird. Auf den Begriff *Search* wurde bewusst verzichtet, um Localina klarer von einem reinen Restaurant-Verzeichnis abzugrenzen. Eine Suche assoziiert nämlich, dass man lediglich auf einen Suchbegriff passende Resultate erhält. Bei Localina geht es jedoch darum, das Restaurant zu *finden*, das am besten dem persönlichen Geschmack entspricht.

Infobox nach dem ersten Start der Applikation

Nach dem ersten Start der Applikation erscheint eine Infobox, welche die Funktionsweise von Localina kurz erklärt (ähnlich wie bei Tellmewhere).

Map-Pin als Icon für Find Locations

Statt der Lupe, wird das Symbol für einen Google Maps-Pin als Tab-Icon für das Tab *Find Locations* verwendet, weil die Lupe schlecht zu der Kartenansicht passt.

Map-Pins

Die Form des Map-Pins wurde geändert, so dass diese die gleiche Form haben wie das Tab-Icon.

Register mit Anzahl der bereits bewerteten Restaurants

Damit der Benutzer sieht, dass die bewerteten Restaurants in das Register *Already Rated* wandern, zeigt das Register die Anzahl Restaurants in der Liste neben dem Titel des Tabs an (vgl. Seite 12 im PDF).

3.2.d Mockup Version 2.2

Die Mockup Version 2.2 stellt die Finale Version des Externen Designs dar. Das vollständige Mockup findet sich unter *Mockups/Mockup-v2.2.pdf* auf der Abgabe-CD. Die wichtigsten Änderungen zu der vorherigen Version werden nachfolgend zusammengefasst.

Location in Place umbenannt

Bei Localina steht der Begriff *Location* für eine Lokalität. Im Englischen ist dieser Begriff jedoch verwirrend, weil eine Location eher einen geografischen Standort bezeichnet. Daher verwendet die Localina iPhone App den Begriff *Place*, welcher am ehesten die Englische Übersetzung für Lokalität ist.

Favorites

Es wurde festgestellt, dass es eine Möglichkeit braucht, ein Bookmark für ein Restaurant setzen zu können, dessen Informationen man sich zu einem späteren Zeitpunkt wieder ansehen möchte. Zu diesem Zweck wurde die Rubrik *Favorites* eingeführt.

My Profile

Die Rubrik *Info* wurde in *My Profile* geändert und informiert den User über seinen Profilstatus (vgl. Seite 16 im PDF). In dieser Ansicht kann er sehen, wie gut er den Empfehlungsalgorithmus bereits trainiert hat (vgl. *Personalization level* im PDF).

Schön wäre, wenn sich der User hier noch ansehen könnte, wie Localina ihn anhand von Adjektiven einschätzt (z.B. romantisch, asiatisch, stylisch,…). Diese könnten wie im Entwurf dargestellt als Tag-Cloud illustriert werden. Diese Anzeige wäre natürlich sehr interessant für den Benutzer, kann allerdings nur in die iPhone App integriert werden, wenn Astina eine entsprechende Funktion über die SOAP-Schnittstelle zur Verfügung stellt.

Vereinfachung der Icons

Die vielen verschiedenen Icons führten eher zu Verwirrung, als dass sie der besseren Darstellung dienten (vgl. Seite 2 im PDF).

Im aktuellen Entwurf gibt es nur noch ein Icon für Empfehlungen:



Restaurants, welche man selbst bewertet hat, werden mit dem Zahlenwert der eigenen Bewertung dargestellt. Alle übrigen Restaurants zeigen die durchschnittliche Bewertung aller User.

Call-Button

Falls die Telefonnummer des Restaurants bekannt ist, erscheint oben rechts ein Call-Button (vgl. Abbildung Seite 9 im PDF).

Popup für Rating

Die Bewertung geschieht nicht mehr direkt in der Detailansicht, sondern in einem Popup (vgl. Seite 2 im PDF). Dadurch wird die Detailansicht übersichtlich und es bleibt mehr Platz für Informationen.

Roter Lösch-Button statt "Ignore Selected Ratings"-Button

Die *Ignore Selected Ratings*-Funktion wurde ersetzt durch einen einfachen Löschknopf neben jedem Restaurant, mit welchem man ein Restaurant aus der Liste entfernen kann ohne es bewerten zu müssen (vgl. Abbildung 14 im PDF).

4 Usability Tests

4.1 Testen der Mockups

Die Mockup-Versionen um Feedback für die weitere Entwicklung der Mockups zu erhalten.

Versuchsanordnung

Die Usability-Tests wurden in einem Sitzungsraum der Astina AG durchgeführt. Bei den Interviews wurden Tonaufnahmen gemacht.

Test-Dokumentation

Die Tests wurden in einem Test-Protokoll dokumentiert. Diese Dokumente finden sich im Verzeichnis *UsabilityTests/MockupTests* auf der Abgabe-CD. Nachfolgend findet sich eine Zusammenstellung der zentralen Aussagen und Beobachtungen.

4.1.a Test der Version 1

Benutzerprofil

- Jurist.
- Männlich, vergeben, 30.
- Technisch interessiert und versiert.
- Besitzt seit rund einem Jahr ein iPhone.

Einführung

"Eine Freundin hat dir von Localina erzählt, einer Applikation die dir Empfehlungen für Restaurants in Zürich aufgrund deines persönlichen Geschmacks geben kann. Du hast diese Applikation von iTunes Store heruntergeladen."

Situation

"Du gehst an deinem Arbeitsplatz immer etwa in die gleichen Restaurants und du würdest gerne etwas Neues kennenlernen. Stell dir vor, du möchtest Empfehlungen in der Nähe deines Arbeitsplatzes, an welchem du dich gerade befindest."

Zentrale Aussagen und Beobachtungen

"Ich möchte eine App, wo ich direkt über OpenTable einen Tisch reservieren kann."

"Ah, super, wenn man die Liste runter skrollt, erscheint ein Button „contact". Damit kann ich das Restaurant sicherlich direkt kontaktieren."

4.1.b Test der Version 2

Benutzerprofil

- Kaufmännische Angestellte.
- Weiblich, Single, 28.
- Technisch mässig interessiert und durchschnittlich versiert.

Einführung

"Eine Freundin hat dir von Localina erzählt, eine Applikation die dir Empfehlungen für Restaurants in Zürich aufgrund deines persönlichen Geschmacks geben kann. Du hast diese Applikation vom iTunes Store heruntergeladen."

Situationen

"Stell dir vor, du möchtest den Shopping-Tag mit deinen Kolleginnen mit einem gemeinsamen Nachtessen in einem für dich neuen Restaurant in Zürich-City abrunden. Wie gehst du vor?"

"Stell dir vor, du warst nun mit deinen Kolleginnen in einem der Restaurants, die dir Localina herausgesucht hat. Am Tag darauf möchtest du nun dieses Restaurant bewerten. Wie würdest du vorgehen?"

Zentrale Aussagen und Beobachtungen

“’Recent cities’, was bedeutet das? War ich dort bereits einmal oder sind dies Städte in meiner Nähe oder wurde bereits mehrmals ein Restaurant in diesen Städten gesucht?”

“[Betrachtet die Suchvorschläge] "Die sind ja gar nicht alphabetisch geordnet, müsste das nicht alphabetisch sein?"

"Mir werden hier also irgendwelche Restaurants aufgezeigt, welche ich kennen könnte. Wie die aber ausgewählt werden, ist mir unklar.”

Interviewer: "Wie gehst du vor, wenn du nun das angezeigte Restaurant „Angkor“ bewerten möchtest?"

Testperson: [überlegt] "Muss ich zuerst auf diesen Kreis vor dem Namen des Restaurants drücken, um bewerten zu können oder kann ich direkt auf das entsprechende Smiley drücken?"

4.2 Release 1

Das externe Design wurde nach Abschluss des Redesigns anhand eines abschliessenden Usability-Tests mit vier Benutzern validiert, um die Qualität des externen Designs sicherzustellen.

Die Versuchspersonen wurden nach folgenden Kriterien ausgewählt:

- Besitzt ein iPhone.
- Geht oft auswärts Essen.
- Wohnt in der Stadt.

Versuchsanordnung

Der Usability-Tests wurden in einem Sitzungsraum der Astina AG durchgeführt. Bei den Interviews wurden Tonaufnahmen gemacht. Ergänzend wurden, falls sinnvoll, folgende Notizen gemacht:

- Original-Zitate.
- Aktionen, Schritt für Schritt.
- Gefühle des Benutzers.

Versuchsablauf

- Willkommen.
- Ausgangslage erklären.
- Auf Aufnahme hinweisen.
- Vertraulichkeit garantieren.
- Profil aufnehmen.
- **Think Aloud** erklären.
- Eine oder zwei konrete Aufgabenstellungen.

Test-Dokumentation

Die Anwendungssituationen, aus welchen am meisten Informationen gewonnen werden konnten, sind in einem Test-Protokoll detailliert beschrieben. Diese Dokumente finden sich im Verzeichnis *UsabilityTests/UsabilityTest2* auf der Abgabe-CD. Nachfolgend findet sich eine Zusammenstellung der zentralen Aussagen und Beobachtungen.

4.2.a Interview User 1

Benutzerprofil

- Männlich, vergeben, 35.
- Technisch durchschnittlich interessiert und versiert.
- Besitzt seit rund zwei Monaten ein iPhone.

Einführung

“Auf Empfehlung eines Freundes hast du dir vor einigen Tagen aus dem AppStore Localina heruntergeladen. Aufgrund der Beschreibung im AppStore und den Erzählungen deines Freundes weisst du, dass dir die App Empfehlungen zu Restaurants aufgrund deines persönlichen Geschmacks geben kann. Du bist auch darüber informiert, dass du selbst Restaurants bewerten musst, um deinem Geschmack möglichst genau entsprechende Restaurants empfohlen zu bekommen. Das heisst, umso mehr Restaurants du bewertest, umso besser werden die Empfehlungen von Localina.”

Situation

"Es ist nun bereits einige Tage her, seit du dir Localina vom AppStore auf dein iPhone geladen hast. Mit der Suchfunktion hast du dich bereits etwas vertraut gemacht. Als du erneut eine Suchanfrage starten willst, entdeckst du neben dem Localina-Icon eine rote Nummer. Was denkst du, bedeutet diese Nummer und wie würdest du vorgehen, wenn du dir darunter noch nichts Genaues vorstellen kannst, jedoch herausfinden möchtest, was es mit dieser Ziffer auf sich hat?“

Zentrale Aussagen und Beobachtungen

“Die selbe Ziffernanzeige finde ich auch oberhalb des Tabs 'Rate Places'. Da ich selbst Restaurants bewerten muss, damit mir Localina meinem Geschmack entsprechende Empfehlungen abgeben kann, werde ich mit dieser Anzeige wohl zum Raten von Restaurants aufgefordert.”

“Also, unter jedem Restaurant gibt einerseits fünf Buttons mit Ziffern von 1 bis 5, andererseits einen roten Button mit Kreuz. Da mir das erste Restaurant sehr gut gefällt, drücke ich auf den Button mit der Ziffer 5.” [Testperson drückt auf den Button mit der Ziffer 5.] “Hmm, nun ist das Restaurant verschwunden.” [Testperson denkt nach.] “Ah, nun hat es beim Register 'Already Rated' eine 1 gegeben. Das vorher von mir bewertete Restaurant wurde also in dieses Register verschoben.”

4.2.b Interview User 2

Benutzerprofil

- Weiblich, vergeben, 29.
- Technisch mässig interessiert und versiert.
- Besitzt seit rund einem halben Jahr ein iPhone.

Einführung

“Eine Freundin hat dir von Localina erzählt, eine Applikation die dir Empfehlungen für Restaurants in Zürich aufgrund deines persönlichen Geschmacks geben kann. Du hast diese Applikation vom iTunes Store heruntergeladen.”

Situation

“Du besuchst heute Abend mit deinem Freund ein Konzert im Club X-tra an der Limmatstrasse 118 in Zürich. Vorher möchtet ihr gerne noch etwas essen gehen. Am liebsten in der Nähe des Clubs.”

Zentrale Aussagen und Beobachtungen

[Testperson liest den Text auf der Infobox.] “Ich spreche leider nicht so gut Englisch.”

[Empfehlungen werden geladen.] “Ooh, da werden mir aber einige Restaurants angezeigt.“

“Ich sehe mehrere grüne Pins. Was bedeuten die schon wieder? Kann ich das Fenster, das zu Beginn erschienen ist, nochmals aufrufen?“

4.2.c Interview User 3

Benutzerprofil

- Männlich, verheiratet, 55.
- Jurist.
- Technisch sehr interessiert, aber mässig versiert.
- Besitzt erst seit rund einem Monat ein iPhone.

Einführung

“*Ein Arbeitskollege hat dir von Localina erzählt, eine Applikation die dir Empfehlungen für Restaurants in Zürich aufgrund deines persönlichen Geschmacks geben kann. Du hast diese Applikation nun vor einigen Tagen vom iTunes Store heruntergeladen, bisher jedoch noch nie benutzt. Du weisst einzig, dass dir die Applikation Empfehlungen zu Restaurants aufgrund deines persönlichen Geschmacks abgeben kann, wenn du selbst genügend Restaurants bewertest.****”***

Situation

"Du gehst an deinem Arbeitsplatz immer etwa in die gleichen Restaurants und würdest deswegen gerne etwas Neues kennenlernen. Stell dir vor, du möchtest Empfehlungen in der Nähe deines Arbeitsplatzes, an welchem du dich gerade befindest.“

Zentrale Aussagen und Beobachtungen

[Die Testperson drückt den Button 'Nearby'.] “Nun erscheint ein Kartenausschnitt. Komisch, es erscheinen aber keine Pins. Muss dies so sein?”

“Hmm, was bedeutet die Farbe der Pins schon wieder? Kann ich wieder zurück zu dieser Erklärung gehen?“

[Such nach dem Info-Button.] "Ah, da oben rechts. Dieser Knopf dürfte ein bisschen auffälliger sein."

“Ich sehe einen Kartenausschnitt mit dem Restaurant-Pin drauf. Kann ich diesen anklicken, um noch einmal zu schauen, wo sich das Restaurant genau befindet?“

“Ah, wenn ich den Stern anklicke, dann erscheint beim Tab 'Favorites' eine Ziffernanzeige mit einer 1 drin. Mit dem Stern-Button kann ich also wahrscheinlich ein Restaurant zu meiner Favoritenliste hinzufügen.”

4.2.d Interview User 4

Benutzerprofil

- Männlich, Single, 40.
- Informatiker.
- Technisch sehr interessiert und versiert.
- Besitzt seit rund eineinhalb Jahren ein iPhone.

Einführung

“Beim Durchstöbern des AppStores bist du per Zufall auf Localina gestossen. Da du gerne dir noch unbekannte Restaurants besuchst, hat die dort aufgeführte Beschreibung über Localina dein Interesse geweckt und du hast dir die App auf dein iPhone geladen.”

Situation

"Dieses Jahr bist du für das Organisieren des alljährlichen Geschäftsessens zuständig. Zu diesem Zweck möchtest du gerade einmal testen, ob dir Localina bei deiner Suche behilflich sein kann. Du hast dir für diesen Abend ein Restaurant in Zürichsee-Nähe vorgestellt. Wie gehst du vor?"

Zentrale Aussagen und Beobachtungen

[Testperson drückt den Button 'By Address', gibt sodann im Suchfeld 'Bellevue' ein und betätigt den blauen Such-Button.] "Uuups, da gibt es wohl noch ein Bellevue in den USA. Da muss ich wohl zusätzlich noch 'Zürich' im Suchfeld eingeben. So funktioniert dies zumindest bei der SBB-App, wenn man eine Tramhaltestelle als Start- oder Endpunkt eingeben will."

[User hat nach Restaurants in der Umgebung des Bellevue gesucht.] "Ausser einem Kartenausschnitt sehe ich nichts." [Testperson denkt nach und schaut sich auf dem Screen um.] "Hmm, ich sehe noch zwei weitere Register. Vielleicht werden mir Restaurants unter einem anderen Register angezeigt." [Testperson klickt das Register 'All in Area' an.] "Super, jetzt werden mir diverse Restaurants angezeigt."

“Wieso steht hier zur Art der Küche und dem Preis nichts? Dies würde mich am meisten interessieren."

Interviewer: "Was verstehst du unter dem Tab 'My Profile'?"

Testperson: "Hmm, das ist eine gute Frage. Vielleicht wird mir darin aufgezeigt, welche Art von Restaurants meinem Geschmack entspricht oder aber wie viele Restaurants ich bereits bewertet habe. Am besten wähle ich einmal den Tab 'My Profile' an." [Testperson wählt den Tab 'My Profile'.] "Okey, hier steht 'Localinas Learning Progress' und 'Here's your opportunity to get personalized recommendations: Rate the places you know and Localina tells you where to go!'. Localina fordert mich hier also auf, noch Restaurants zu bewerten."

5 Internes Design

5.1 Domainmodell

5.1.a Strukturdiagramm

Abbildung II-1 zeigt das Domainmodell der Localina iPhone App.

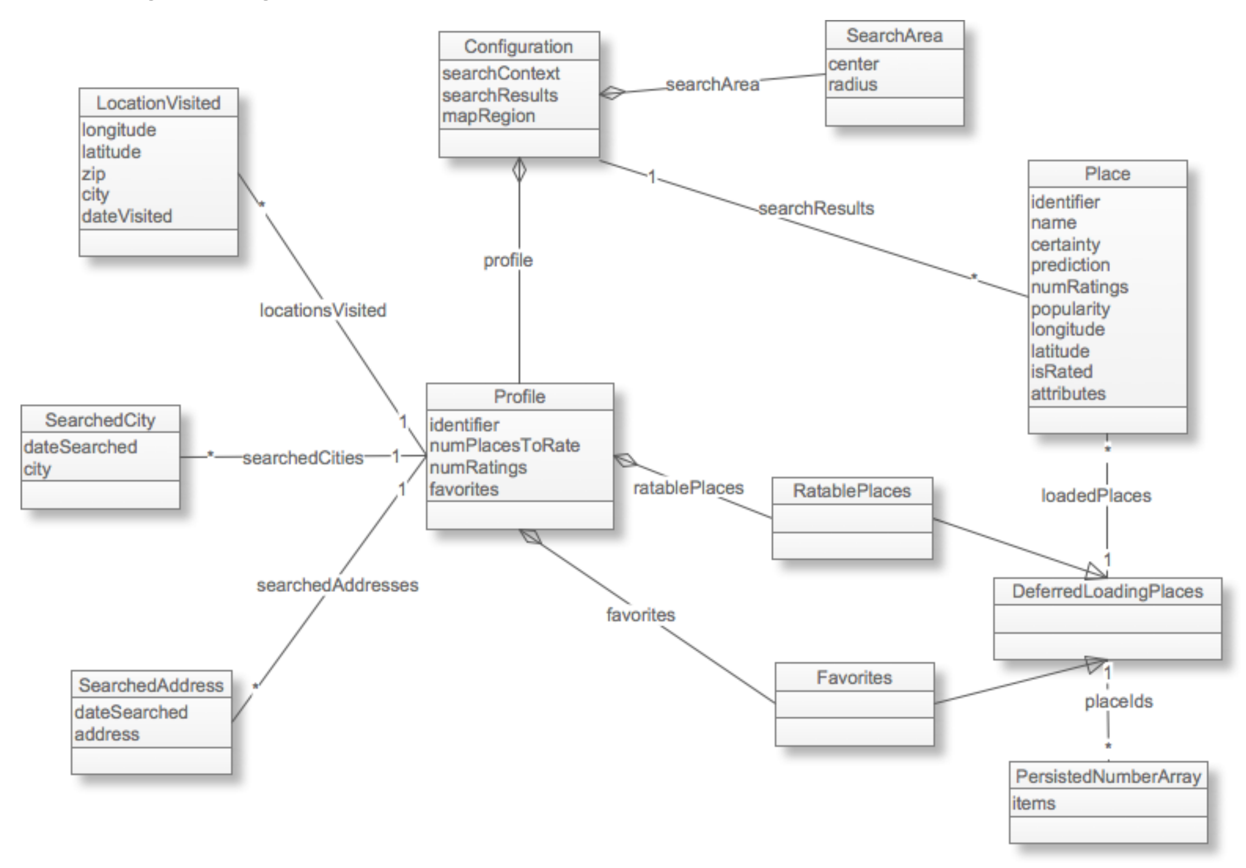


Abbildung II-1: Domainmodell der Localina iPhone App

5.2 Klassenkonzepte

In diesem Kapitel werden die Konzepte der Klassen des Domainmodells erläutert.

5.2.a Configuration-Konzept

Diagramm

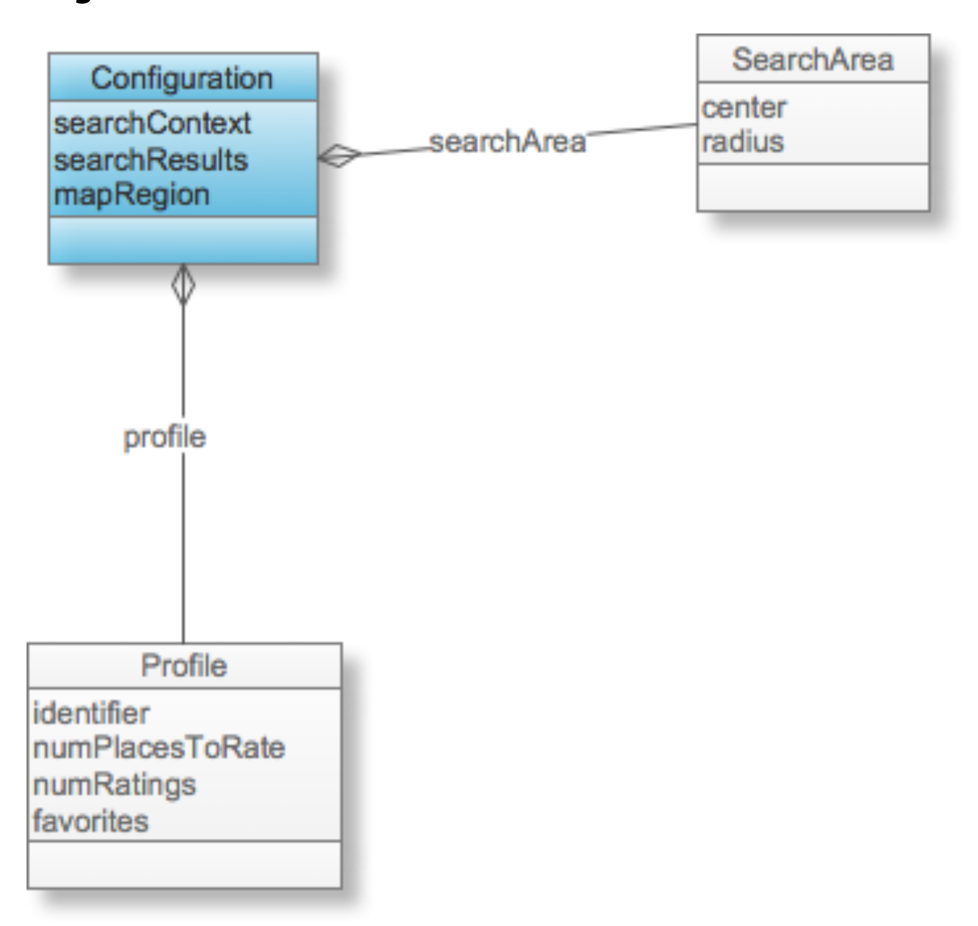


Abbildung II-2: Configuration-Konzept

Beschreibung

Jede Instanz der Localina iPhone App besitzt genau ein Configuration-Objekt, welches den Zustand der Applikation speichert. Die Configuration besitzt auch eine Referenz zum Profile-Objekt, welches Angaben zum Benutzerprofil speichert.

Attribute

Name	Beschreibung	Wertebereich
searchContext	Aktueller Suchkontext des Benutzers (selektiertes Register in der Resultat-View).	ENUM (SearchContextFindMyTaste, SearchContextFindPopularPlaces, SearchContextFindAll)
searchArea	Umkreis der letzten Suche.	SearchArea
searchResults	Resultate der letzten Suchanfrage.	Array mit Instanzen von Place
mapRegion	Speichert den zuletzt gewählten Kartenausschnitt.	MKMapRegion
profile	Referenz auf das aktuelle Benutzerprofil.	Profile

Beziehungen

Eine Configuration **verwaltet genau ein Profile (Profile-Konzept) und eine SearchArea (**

SearchArea-Konzept). Es wäre auch eine Implementation denkbar, wo die Configuration mehrere Profile verwaltet (z.B. um mehrere Moods abzubilden).

5.2.b Profile-Konzept

Diagramm

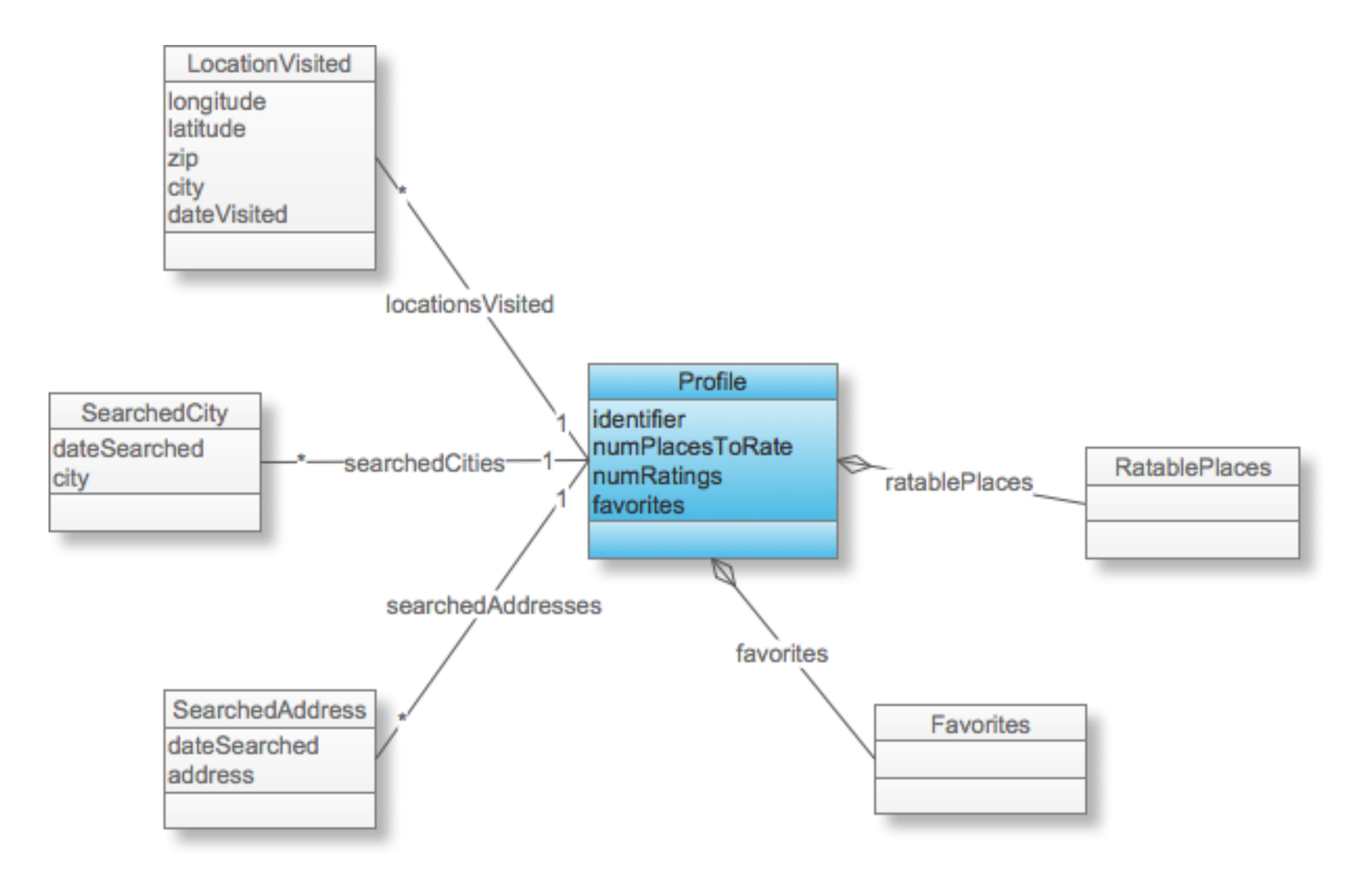


Abbildung II-3: Profile-Konzept

Beschreibung

Das **Profile** beinhaltet alle Daten des Benutzerprofils. Theoretisch kann ein Benutzer mehrere Profile haben, z.B. um wechselnde Präferenzen (Moods) abzubilden (Geschäftsessen, romantisches Dinner mit der Partnerin/dem Partner, Bier mit Freunden). Die Localina iPhone App besitzt aber nur ein Profil, weil sich gezeigt hat, dass das Konzept mit mehreren Moods den Benutzer mehr verwirrt, als es einen Vorteil bringt.

Attribute

Name	Beschreibung	Wertebereich
identifier	Beliebiger String, welcher den Benutzer bei der Localina Application eindeutig identifiziert und unter welchem die Benutzerratings serverseitig gespeichert werden. Der Identifier wird beim ersten Applikationsstart automatisch aus der Geräte-ID und dem Timestamp generiert.	String
numPlacesToRate	Speichert die Anzahl der Rating-Tasks und wird in den NSUserDefaults persistiert.	Ganzzahl
numRatings	Speichert die Anzahl der Bewertungen, welche der User abgegeben hat.	Ganzzahl
ratablePlaces	Verwaltet die noch zu ratenden und die bereits gerateten Places. Die Localina iPhone App persistiert nur die IDs der Places. Die übrigen Attribute werden erst geladen, wenn sie benötigt werden (vgl. DeferredLoadingPlaces-Konzept).	RatablePlaces
favorites	Verwaltet die Places, welche der Benutzer zu den Favoriten hinzugefügt hat. Die Localina iPhone App persistiert nur die IDs der Places. Die übrigen Attribute werden geladen, wenn sie benötigt werden (vgl. DeferredLoadingPlaces-Konzept).	Favorites

5.2.c Place-Konzept

Diagramm

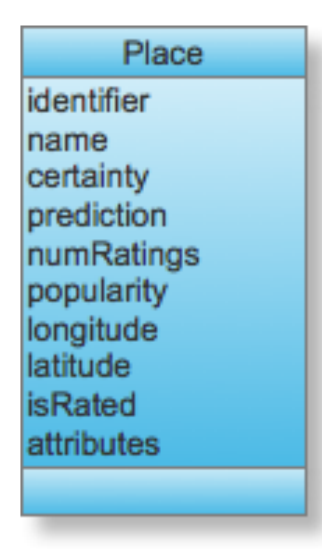


Abbildung II-4: Place-Konzept

Beschreibung

Das Konzept **Place** beschreibt eine Lokalität mit einer Predition. Bei Localina wird eine Lokalität durch ein Location-Objekt beschrieben, welches einem separaten Prediction-Objekt angehängt ist. Weil eine Lokalität der Localina iPhone App jedoch nur eine Prediction haben kann (Einzelbenutzer-Anwendung), werden die Attribute der Prediction direkt dem Place-Objekt angehängt.

Attribute

Name	Beschreibung	Wertebereich
identifier	Eindeutige ID, unter welcher die Lokalität in der Localina Datenbank referenziert wird.	Ganzzahl
name	Name der Lokalität.	String
certainty	Zuverlässigkeit/Genauigkeit der Prediction für die Lokalität.	-1 < x < 1
prediction	**Prediction** für die Lokalität.	-1 < x < 1
numRatings	Gesamtzahl der Bewertungen, welche für die Lokalität abgegeben wurden.	Ganzzahl
popularity	Durchschnittliche Bewertung der Lokalität.	-1 < x < 1
longitude	Längengrad des Standorts der Lokalität.	0.0 - ±90.0
latitude	Breitengrad des Standorts der Lokalität.	0.0 - ±180.0
isRated	Gibt an, ob der Benutzer die Lokalität bereits bewertet hat.	BOOLEAN

5.2.d SearchArea-Konzept

Diagramm

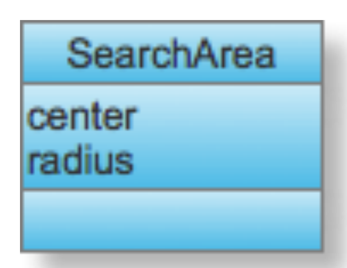


Abbildung II-5: SearchArea-Konzept

Beschreibung

Das Konzept **SearchArea** beschreibt den Umkreis der letzten Suchanfrage.

Attribute

Name	Beschreibung	Wertebereich
center	Ort der Suche.	Geokoordinate
radius	Suchradius in Kilometer.	Ganzzahl

5.2.e LocationVisited-Konzept

Diagramm

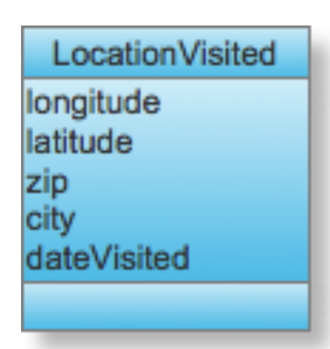


Abbildung II-6: LocationVisited-Konzept

Beschreibung

LocationVisited beschreibt einen Benutzerstandort, wo sich der Benutzer mittels LBS des iPhone geortet hat (jedes Mal, wenn er die Funktion *Find Restaurants Nearby* wählt).

Attribute

Name	Beschreibung	Wertebereich
longitude	Längengrad des Standorts.	0.0 - ±90.0
latitude	Breitengrad des Standorts.	0.0 - ±180.0
zip	Postleitzahl des Standorts.	String
city	Stadt, in der sich der Benutzer während der Lokalisierung befunden hat.	String
dateVisited	Datum der Lokalisierung.	Datum

5.2.f SearchedCity-Konzept

Diagramm

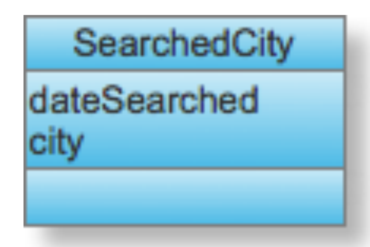


Abbildung II-7: SearchedCity-Konzept

Beschreibung

Das Konzept **SearchedCity** beschreibt eine Stadt, nach welcher der Benutzer bereits Empfehlungen gesucht hat.

Attribute

Name	Beschreibung	Wertebereich
dateSearched	Datum der Suchanfrage.	Datum
city	Stadt, nach welcher der Benutzer gesucht hat.	String

5.2.g SearchedAddress-Konzept

Diagramm

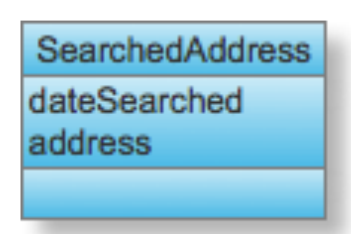


Abbildung II-8: SearchedAddress-Konzept

Beschreibung

Das Konzept *SearchedAddress* beschreibt eine Adresse, nach welcher der Benutzer bereits unter *Find Restaurants by Address* gesucht hat. Alle Suchanfragen werden als SearchedAddress-Objekt gespeichert und die 5 zuletzt gesuchten Adressen werden unter *Recent Searches* aufgelistet.

Attribute

Name	Beschreibung	Wertebereich
dateSearched	Datum der Suchanfrage.	Timestamp
address	Adresse, nach welcher der Benutzer gesucht hat. Die Adresse wird mittels Google Geocoding formatiert.	String

5.2.h DeferredLoadingPlaces-Konzept

Diagramm

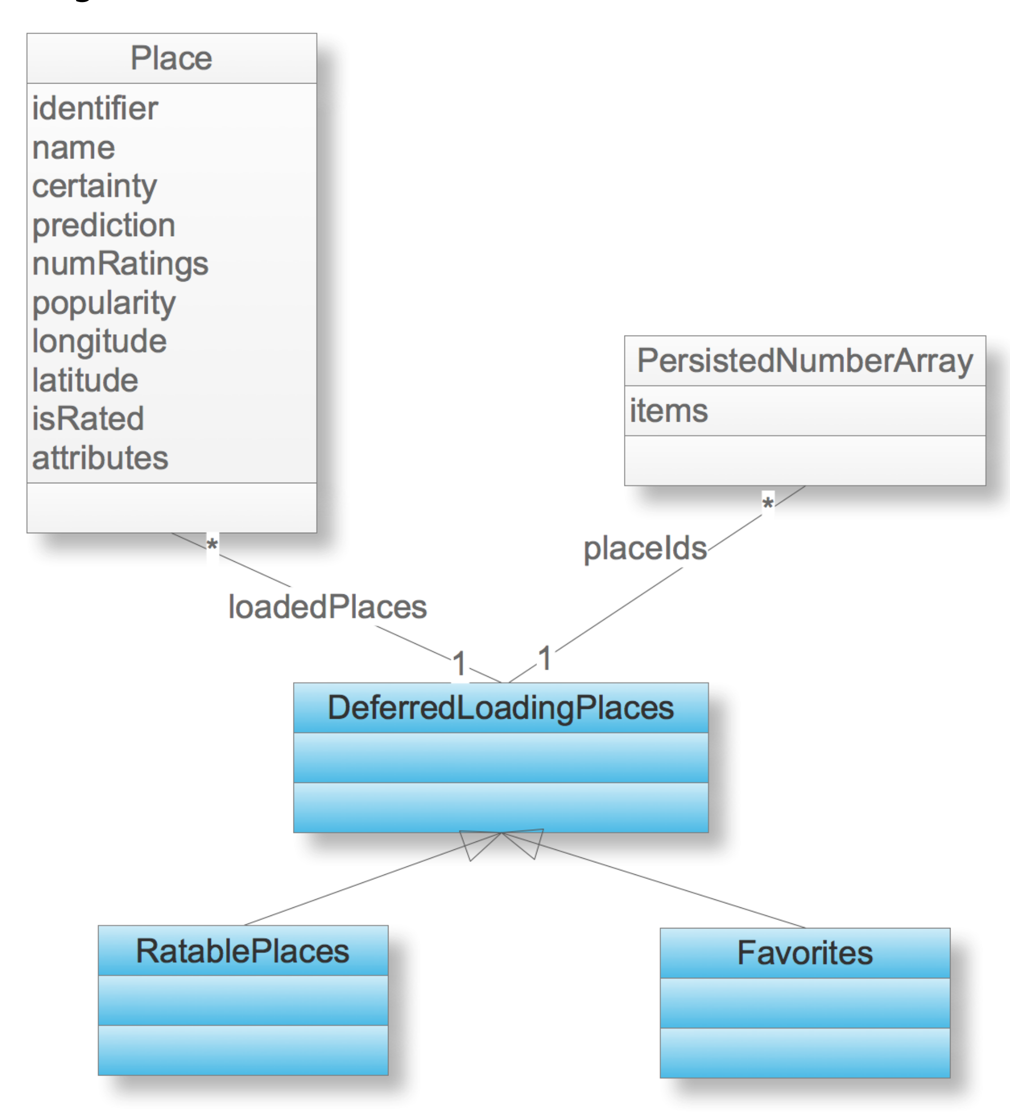


Abbildung II-9: DeferredLoadingPlaces-Konzept

Beschreibung

DeferredLoadingPlaces verwaltet Objekte, welche erst bei Bedarf vom Server geladen werden. Es verwaltet eine Liste von allen Place-IDs und eine separate Liste mit allen geladenen Place-Objekten. Das Konzept wird bei den RatablePlaces (Liste aller Places unter *Places to Rate* und *Already Rated Places*) und bei den Favorites (Places, welche der Benutzer zur Favoritenliste hinzugefügt hat) angewendet. Die Liste der Favorites und der

RatablePlaces wird nämlich auf dem iPhone persistiert. Weil Localina ein sehr dynamisches System ist und sich die Attribute von Place ständig ändern, werden nur die IDs persitiert. Die Place-Objekte werden bei Bedarf vom Server angefordert. Durch das Deferred Loading sind beim Start der Applikation weniger Requests nötig und wenn der Benutzer zu Beginn gleich Empfehlungen abfragen möchte, stehen diese schneller zur Verfügung.

Attribute

Name	Beschreibung	Wertebereich
loadedPlaces	Array mit allen Place-Objekten, die bereits vom Server geladen worden sind.	Places
placeIds	Adresse, nach welcher der Benutzer gesucht hat. Die Adresse wird mittels Google Geocoding formatiert.	PersistedNumberArray

5.2.i PersistedNumberArray-Konzept

Diagramm

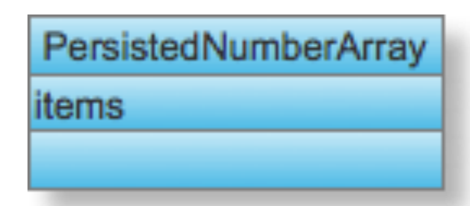


Abbildung II-10: PersistedNumberArray-Konzept

Beschreibung

Das Konzept **PersistedNumberArray** erlaubt es, einen Array mit Number-Objekten mittels Core Data zu persistieren. Es wird überall dort benötigt, wo eine Liste mit Ganzzahlen im Persistent Store gespeichert wird (RatablePlaces, Favorites).

Attribute

Name	Beschreibung	Wertebereich
items	Liste der vom Managed Object Context von Core Data verwalteten Objekte.	Array mit Ganzzahlen

5.3 Architektur

5.3.a Übersicht

Abbildung II-11: Architektur zeigt die Architektur der Localina iPhone Applikation. Der Grundaufbau erfolgt nach dem Muster des Model View Controller-Pattern (MVC). Der Code ist in folgende Pakete gegliedert. Place+SOAP Factory ist eine **Category**, welche die Klasse Place um einen Kostruktor erweitert, der ein Place-Objekt aus einem SOAP-XML erzeugen kann. Place+SOAP stellt die einzige Kopplung zwischen der Communication und dem Model dar. Nachfolgend werden alle Pakete kurz erläutert. Auf die wichtigsten Pakete wird im Abschnitt Wichtigste Pakete genauer eingegangen.

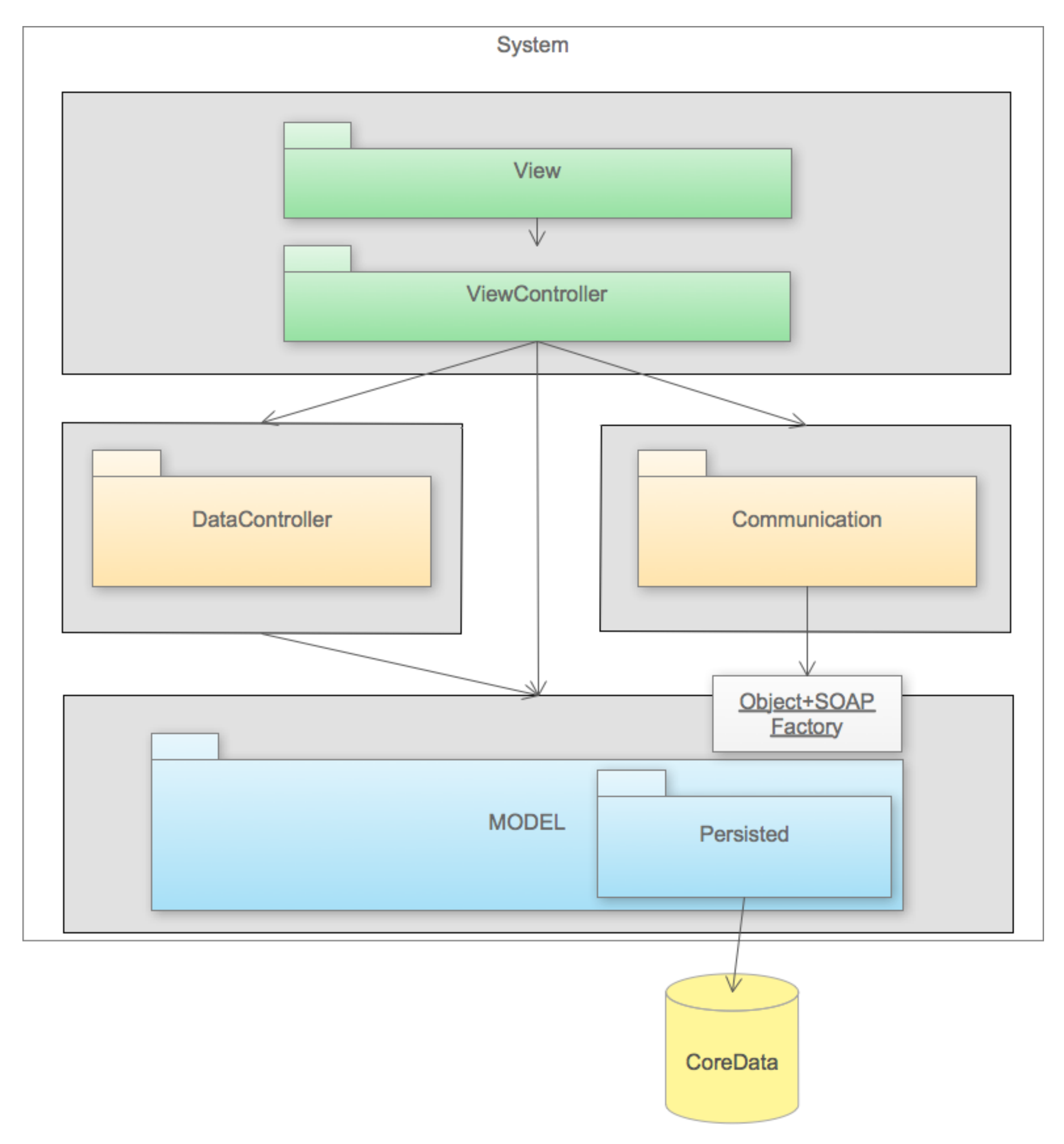


Abbildung II-11: Architektur der Localina iPhone App

Model

In diesem Paket befinden sich alle Klassen, welche die Objekte der Problem Domain beschreiben. Objekte, welche vom DataStore verwaltet werden, werden im Unterpaket *Persisted* zusammengefasst.

Views

Im Paket *Views* befinden sich alle Klassen, welche ein GUI-Objekt beschreiben. Sie leiten alle von UIView ab und besitzen keine Applikationslogik. Die Klassen in diesem Paket sind entweder Erweiterungen von Standard-GUI-Elementen (vgl. CustomAlertView) oder stellen vorkonfigurierte Views dar (vgl. LocalinaNavigationView).

ViewController

Die Klassen in diesem Paket haben die Aufgabe des Controllers im MVC-Pattern. Jede View besitzt einen ViewController, welcher die Daten der View verwaltet und über Delegate Methoden Events vom Event-Loop entgegennimmt und verarbeitet.

DataController

Die DataController stellen Methoden für den einfacheren Zugriff auf Model-Objekte und deren Filterung zur Verfügung. Sie werden von den ViewControllern verwendet, um Model-Objekte der View zu verwalten.

Communication

Das Communication-Paket stellt eine interne Klassenbibliothek für die Kommunikation mit der Localina Application dar. Sie implementiert die Services der SOAP-Schnittstelle der Localina Application. Die Klasse CommunicationFacade stellt ein stark abstrahiertes Interface zu den SOAP-Services von Localina zur Verfügung.

Support

Das Paket Support fasst alle Klassen zusammen, welche den Funktionsumfang der vorhandenen Frameworks erweitern.

5.3.b Externe Abhängigkeiten

Die nachfolgende Tabelle zeigt die externen Abhängigkeiten der Localina iPhone App. Abbildung II-12 stellt die externen Abhängigkeiten grafisch dar.

Bibliothek	Version	Beschreibung
libxml2	2.2	Programmbibliothek zum Parsen von XML-Dokumenten.
TouchXML	1.0.8	Schlanke und performante Alternative zu den NSXML-Klassen, welche auf der libxml2-Bibliothek basiert.
SBJSON	2.2	Framework zum Parsen und Generieren von JSON.
UIKit	iPhoneOS3.0.sdk	UI Framework der iPhone Runtime.
Foundation Framework	iPhoneOS3.0.sdk	Stellt einen Basislayer von Objective-C Klassen wie NSObject, NSString, NSArray uvm. zur Verfügung.
Core Graphics	iPhoneOS3.0.sdk	Window-Server (auch bekannt unter Quartz).
Map Kit	iPhoneOS3.0.sdk	Framework, welches den Google Mobile Maps Service für das iPhone integriert.
Core Location	iPhoneOS3.0.sdk	Stellt und einen Service zur Richtungsbestimmung zur Verfügung.
Quartz Core	iPhoneOS3.0.sdk	Für spezielle grafische Effekte und Transformationen.
Core Data	iPhoneOS3.0.sdk	Data Model Framework, welches ebenfalls Funktionen zur Persistierung der Daten bereitstellt.
CFNetwork	iPhoneOS3.0.sdk	Abstrahiert verschiedene Netzwerkprotokolle.

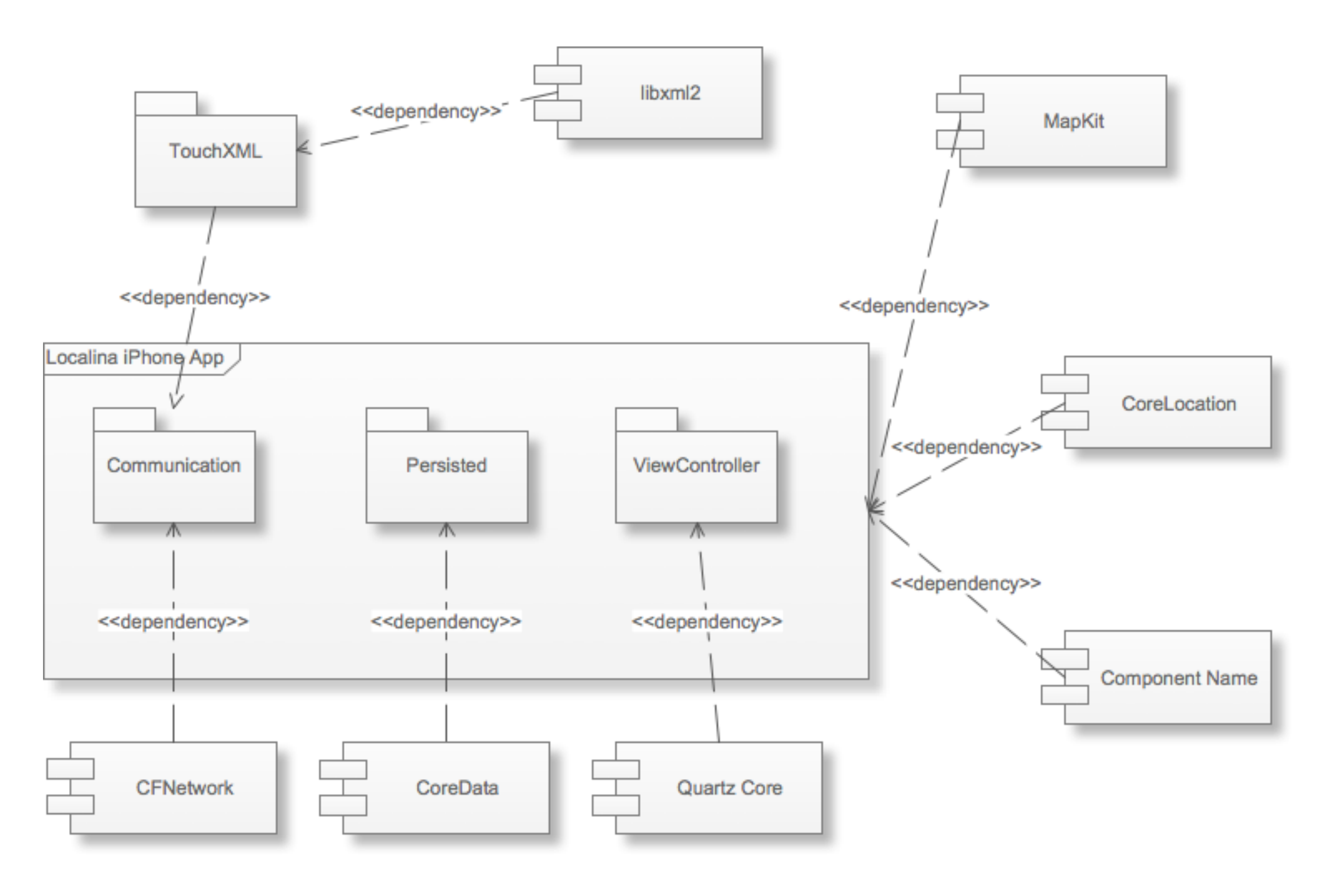


Abbildung II-12: Externe Schnittstelle

5.4 Wichtigste Pakete

In diesem Abschnitt werden die zentralen Pakete der Localina iPhone App sowie deren wichtigste Konzepte und Zusammenhänge nochmals genauer erklärt. Hier werden nur die Klassenmodelle dargestellt, welche zusätzlich erläutert werden. Auf die Darstellung der Attribute und Methoden wurde bewusst verzichtet, um die Übersicht zu verbessern. Wenn man das Projekt in Xcode öffnet, findet man im Ordner des Pakets eine Datei mit dem Klassenmodell, das direkt in Xcode betrachtet und navigiert werden kann. Dort kann man auch die Attribute und Methoden ansehen.

5.4.a Model

Die wichtigsten Klassen des Model wurden bereits im Kapitel *5.2 Klassenkonzepte* beschrieben. Die Darstellung Abbildung II-13: Klassenmodell des Model-Pakets zeigt das Klassenmodell des Model-Pakets. Das innere Paket stellt das Unterpaket *Persisted* dar, welches alle Model-Klassen zusammenfasst, welche Teil des Managed Object Context von Core Data sind. PersistedNumberArray leitet selbst nicht von NSManagedObject ab, gehört aber konzeptionell in dieses Paket, weil seine Elemente als direkte Instanzen von NSManagedObject persistiert werden.

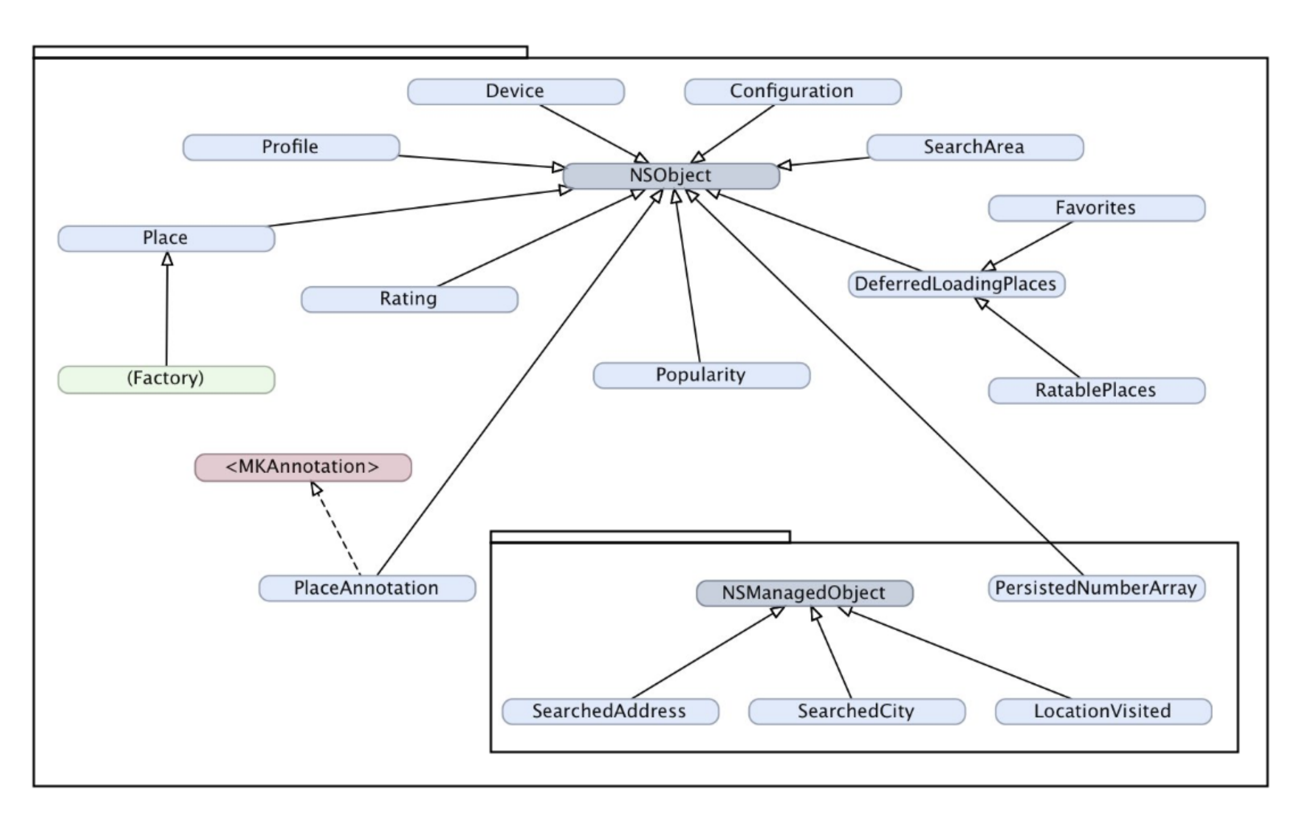


Abbildung II-13: Klassenmodell des Model-Pakets

5.4.b ViewController

Die Abbildung II-14 stellt die Abhängigkeiten und die Hierarchie der *ViewController* dar. Die blauen Objekte sind Instanzen von Klassen, welche in der Localina iPhone App implementiert sind. Die übrigen Objekte sind direkt von Klassen des UIKit abgeleitet.

Zu bemerken ist, dass *favoritesTableViewController* und *ratePlacesViewController* von einem *UINavigationController* verwaltet werden, obwohl sie keine Navigation verwenden. Der Grund ist, dass für diese Views eine Navigation vorgesehen ist, deren Implementation jedoch den Rahmen dieser Arbeit gesprengt hätte.

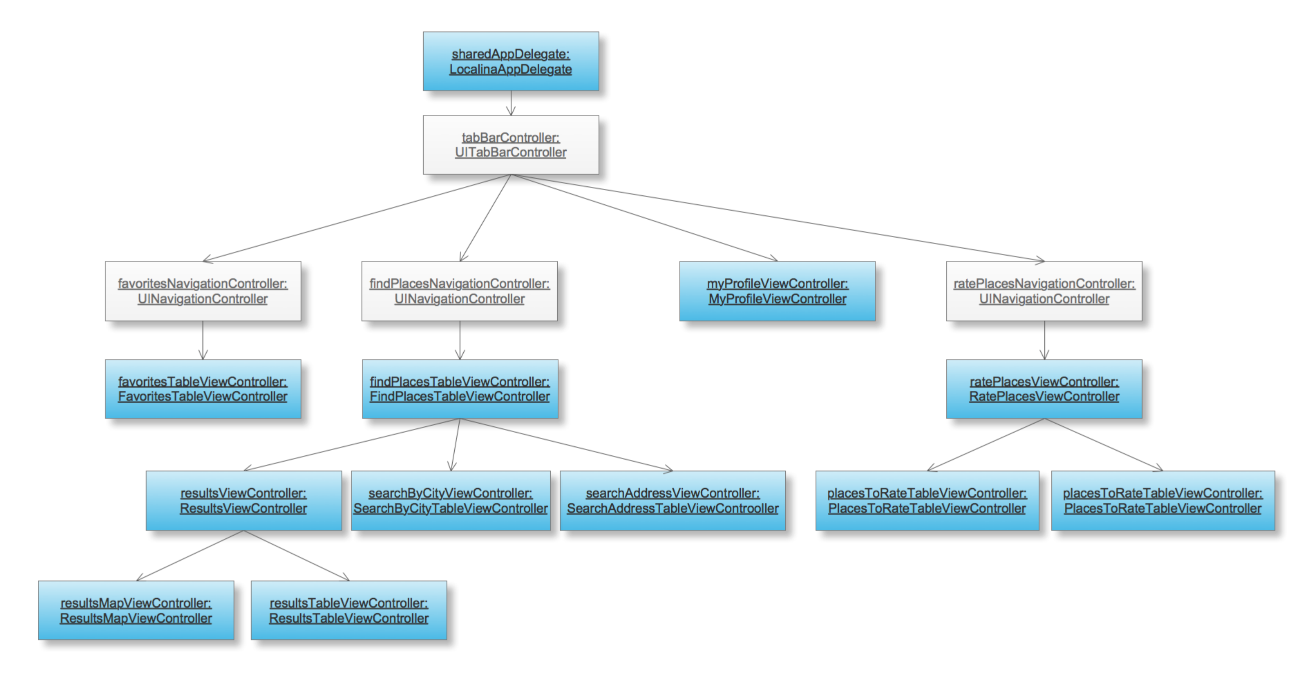


Abbildung II-14: Hierarchie der ViewController

5.4.c DataController

Die Abbildung Abbildung II-15 stellt das Klassenmodell des Pakets *DataController* dar. *PlaceTableDataModel* verwaltet die Place-Objekte, welche von den UITableViewControllern verwendet werden, um diese in der UITableViews darzustellen.

SearchResults stellt statische Methoden zum vereinfachten Zugriff auf die Suchresultate zur Verfügung.

Die *PlaceFilter* implementieren speziell Filter, welche u.a. vom *MapViewController* verwendet werden, um nur die populären Restaurants anzuzeigen, oder nur jene, welche auf den persönlichen Geschmack passen.

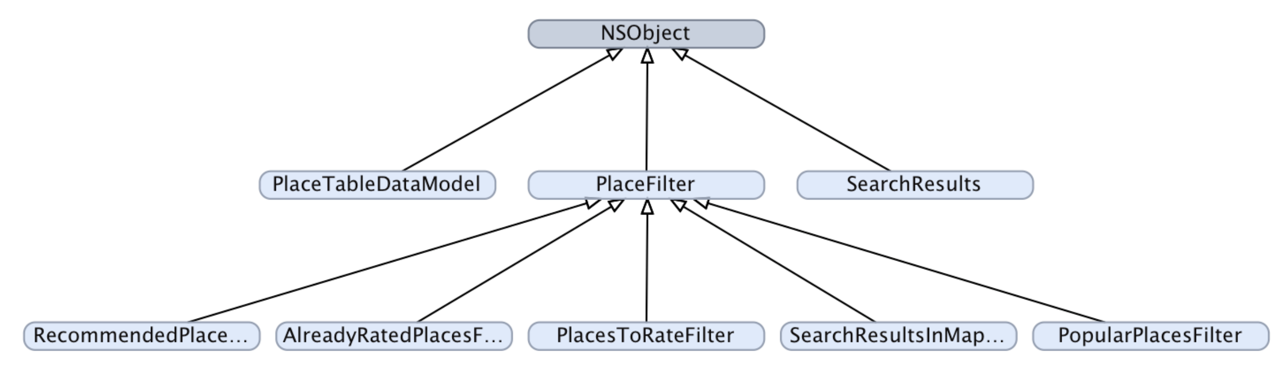


Abbildung II-15: Klassenmodell des DataController-Pakets

5.4.d Communication

Die folgenden Klassendiagramme zeigen, dass das Communication-Paket grob in drei Gruppen von Klassen aufgeteilt werden kann, der *SOAPCommunication* und den *SOAPRequests* bzw. *SOAPResponses.*

Die *SOAPCommunication* verwaltet die Verbindung mit der Localina Application. Ihr wird bei der Initialisierung ein Request- und ein Response-Objekt mitgegeben. *SOAPCommunication* erbt von NSOperation. Dadurch kann die Kommunikation auf einem separaten Thread aufgerufen werden. Bei der Localina iPhone App läuft jede Kommunikation mit der **Localina Application** (aktive Instanz von *SoapCommunication*) asynchron in einem separaten Thread.

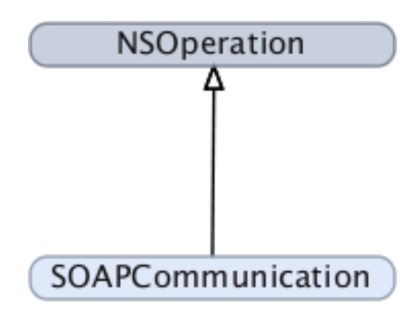


Abbildung II-16: SOAPCommunication-Objekt

SOAPRequest erzeugt aus den Parametern des Request ein SOAP-XML, welches mittels HTTP-Request an die Localina Application übermittelt wird. Für jeden SOAP-Service gibt es eine konkrete Implementation von *SOAPRequest*, welche die Struktur der Anfrage kennt und Methoden zum Setzen der Requestparameter bereitstellt.

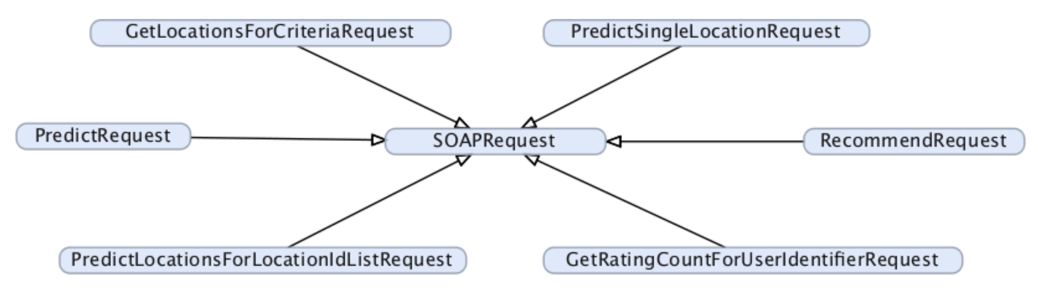


Abbildung II-17: SOAP-Request

SOAPResponse bietet eine Methode zum Parsen der Response der Localina Application. Für jedes Response-Format gibt es eine konkrete Implementation, welche das Response-Format kennt und parsen kann. Diese überschreiben falls nötig auch die Methoden isValid() und isSuccess(), falls eine Response ein nicht standardmässiges Feedback zurückgibt.

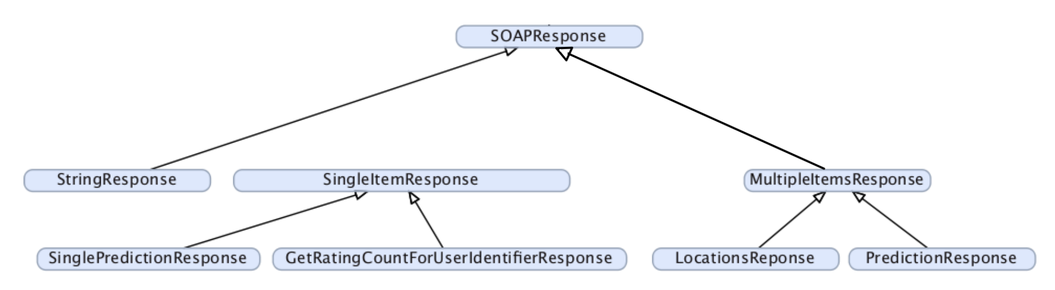


Abbildung II-18: SOAP-Response

CommunicationFacade

Wie Abbildung II-19: CommunicationFacade zeigt, besitzt das Communication-Paket über eine Facade, welche einen einfachen Zugriff auf die SOAP-Services der Localina Application bietet, ohne dass man die Klassen des Communication Pakets kennen muss. Die *CommunicationFacade* gehört jedoch nur logisch zum Communication-Paket. Sie ist bewusst applikationsspezifisch, um ein einfacheres Interface bereitstellen zu können. Falls das Communication-Paket in einer anderen Applikation verwendet wird, muss eine eigene *CommunicationFacade* geschrieben werden.

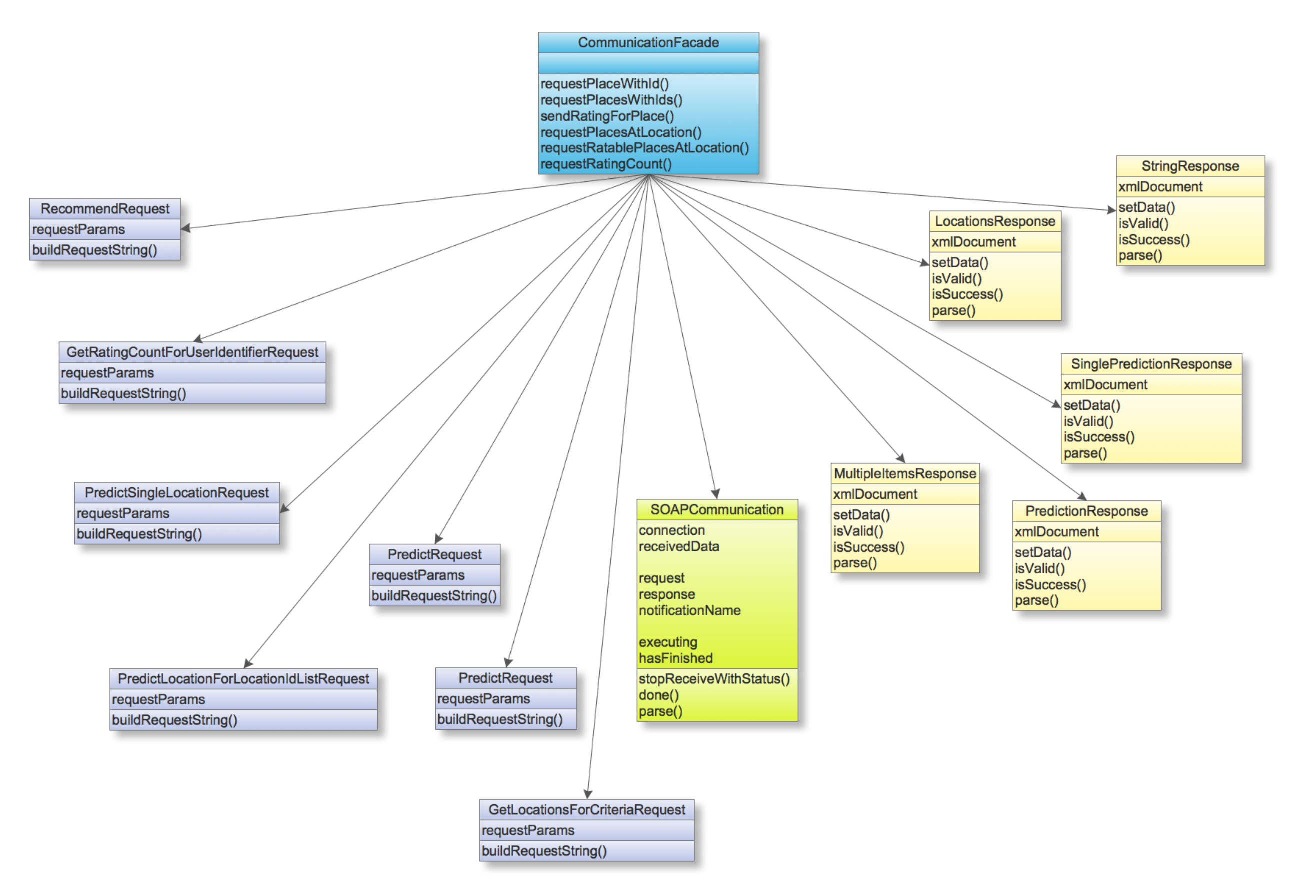


Abbildung II-19: CommunicationFacade

Place+SOAP (Factory)

Place+SOAP ist eine Kategorie (vgl. Categories in Objective-C), welche die Klasse Place um einen Konstruktor erweitert, der ein Place-Objekt aus einem Prediction- oder Location-SOAP-XML erzeugen kann. Place+SOAP gehört daher nur logisch zum Communication-Paket. Falls das Communication-Paket in einer anderen Applikation verwendet wird, muss eine eigene Kategorie geschrieben werden, welche die Struktur des Datenmodells kennt.

6 Implementation

6.1 Wichtige Konzepte

6.1.a Learning Progress

Der Learning Progess soll dem Benutzer ein Feedback geben, wie gut Localina seinen persönlichen Geschmack bereits kennt.



Abbildung II-20: Learning Progress

Der Learning Progress wird aus der Anzahl der Bewertungen berechnet, welche der Benutzer abgegeben hat. Weil das Empfehlungsystem zu Beginn einen grösseren Lernfortschritt erziehlt, als nach einer gewissen Anzahl Bewertungen, folgt die Lernkurve einer logarithmischen Funktion. Folgende Formel wird als gute Näherung angenommen.

100 ∗ (1 – e–0.05∗N)

Diese Formular wurde nicht anhand wissenschaftlicher Methoden geprüft, aber genügt um dem Benutzer ein gutes Feedback zum Lernstatus des Algorithmus geben zu können.

6.1.b Rating Scheduler

Ein Problem von Localina ist, dass die Benutzer in der Situation, wo sie Localina verwenden (um rasch einen Restaurant-Tipp zu erhalten) keine Lust oder Zeit haben, eine Liste von Restaurants zu bewerten. Der Rating Scheduler soll dieses Problem lösen, indem

er dem Benutzer in regelmässigen Abständen Restaurants zum Bewerten vorschlägt, welche der Benutzer mit einer möglichst hohen Wahrscheinlichkeit kennt. Es wurden die folgenden drei Ansätze evaluiert, wie das System ableiten kann, welche Restaurants der User kennt:

(1) Aktionen merken, welche darauf hindeuten, dass der Benutzer das Restaurant nächstens besucht (z.B. Klick auf Call-Button). Wenn der Benutzer ein Restaurant angerufen hat, wird er es ein paar Tage später wahrscheinlich beurteilen können.

(2) Restaurants in der Nähe von Restaurants vorschlagen, welche der User bewertet hat.

(3) Restaurants im Umkreis von Orten vorschlägen, wo sich der Benutzer mittels LBS häufig lokalisiert hat.

Evaluation

Ansatz	Verlässlichkeit	Anzahl potentieller Resultate
1	Sehr hoch	Klein
2	Mittel	Hoch
3	Mittel	Hoch

Ansatz 1 wurde verworfen, weil es zuwenig potentielle Resultate gibt. Aus Ansatz 2 und 3 wurde 3 gewählt, weil er bereits funktioniert, wenn noch keine Restaurants bewertet wurden und weil der Benutzer damit aufgefordert wird, an allen Orten Restaurants zu bewerten, wo er sich auskennt.

Funktionsprinzip des Rating Schedulers

Die Applikation speichert alle Orte, wo sich der Benutzer mittels LBS lokalisiert hat (vgl. LocationVisited-Konzept. Die Funktion des Rating Schedulers wird in der Abbildung II-21: Rating Scheduler in einem Flowchart illustriert. Nachfolgend einige Erläuterungen:

(1) Der Rating Scheduler wir beim Applikationsstart aufgerufen.

(2) Falls seit den letzten Rating-Tasks weniger als 5 Tage vergangen sind, wird der Scheduler beendet.

(3) visitedLocations.shuffle(): Die Liste der Orte, wo sich der Benutzer mittels LBS lokalisiert hat (visitedLocations), wird umsortiert. Das Mischen verhindert, dass immer Restaurants in der Nähe der selben Orte abgefragt werden.

(4) Die Orte in der gemischten Liste werden der Reihe nach verwendet, um vom Server zu bewertende Restaurants in deren Nähe abzufragen. Orte, an welchen sich der User häufiger aufhält, generieren dadurch auch mehr Resultate. Die Suche wird abgebrochen, sobald die Localina Application bereits 5 gültige Vorschläge liefern konnte, bereits 5 Abfragen getätigt wurden oder visitedLocations leer ist. Durch die Beschränkung der Anzahl Abfragen wird verhindert, dass die Internetverbindung nach dem Start der Applikation nicht für eine zu lange Zeit belastet ist.

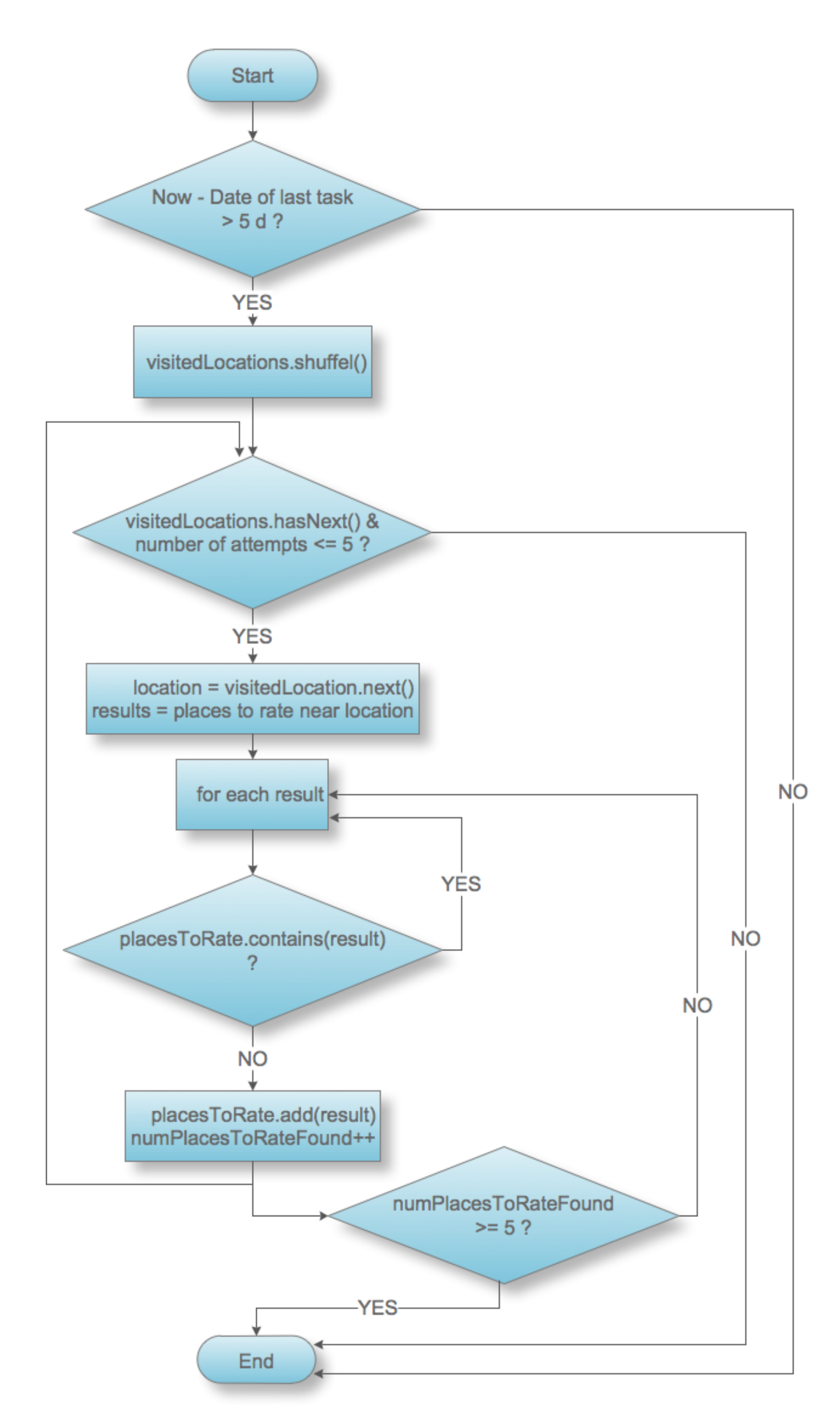


Abbildung II-21: Rating Scheduler

6.2 Automatische Tests

6.2.a Unit-Tests

Die wichtigsten Funktionen der Localina iPhone App wurden anhand von Unit-Tests überprüft. Die Unit-Tests befinden sich im Verzeichnis *Test* des Xcode-Workspace.

Weil die Kommunikation mit der Localina Application asynchron ist, konnten keine Unit-Tests implementiert werden, welche die gesamte Kommunikation testen. Das korrekte Verarbeiten der Response wird daher mit Hilfe einer Mock-Response sichergestellt.

Nachfolgend werden die Unit-Tests kurz erklärt.

TestParsePrediction

Parsed das Resultat der Mock-Response (Standard-Response der Localina Application) und generiert daraus eine Liste von Place-Objekten. Dabei wird geprüft, dass die richtige Anzahl Objekte in der Resultat-Liste vorhanden ist und dass die Place-Objekte die korrekten Attributwerte aus dem XML enthalten.

RatingTests

Die Klasse Rating wandelt den Float-Wert des Ratings in den Rating-Level um, welcher von der Localina iPhone App bei der Anzeige der Pins verwendet wird. Durch die Rating-Klasse wird also bestimmt, welche Bedeutung einem bestimmten Zahlenwert zukommt. Für das Fine-Tuning der Resultatanzeige muss also lediglich das Mapping in der Klasse Rating angepasst werden. RatingTests prüft, dass für jeden möglichen Zahlenwert der erwartete Wert zurückgeliefert wird.

AnnotationTypeTests

Der AnnotationType wird verwendet, um dem Place-Objekt das richtige Icon zuzuordnen.
Diese Klasse überprüft, dass für alle möglichen Werte (RatingLevel, PopularityLevel, Certainty) das richtige Icon bestimmt wird.

PopularityTests

Die Klasse Popularity wandelt den Float-Wert der Popularity in den Popularity-Level um, welcher von der Localina iPhone App bei der Anzeige der Pins verwendet wird. Durch die Popularity-Klasse wird also bestimmt, welche Bedeutung einem bestimmten Zahlenwert zukommt. Für das Fine-Tuning der Resultatanzeige muss also lediglich das Mapping in der Klasse Popularity angepasst werden. PopularityTests prüft, dass für jeden möglichen Zahlenwert der erwartete Wert zurückgeliefert wird.

6.3 Manuelle Tests

6.3.a Applikationsverhalten

Mittels diesen Tests wird die Konsistenz der Applikation überprüft.

Beschreibung	Aktion	Erwartetes Verhalten	Pass/Fail	Bemerkungen
Beim ertmaligen Starten der Applikationen werden Informationen zur Funktionsweise von Localina angezeigt.	Erstmaliges Starten der Applikation.	Infobox erscheint auf dem Display.	✓	
Applikation Badge zeigt den korrekten Wert an.	Applikation starten und die Anzahl der *Places to Rate* zählen. Applikation beenden.	Der Badge auf dem App-Icon muss der Anzahl der *Places to Rate* entsprechen.	✓	
Favoriten werden persistiert.	Lokalität zu den Favoriten hinzufügen und Applikation neu starten.	Die Lokalität muss in der Favoritenliste erschein.	✓	
Gesuchte Adressen werden persistiert.	Nach Adresse suchen und Applikation beenden.	Adresse erscheint als erstes Element in der *Recent Searches*.	✓	

6.3.b Map-Ansicht der Resultate

Mittels diesen Tests wird überprüft, dass in den jeweiligen Ansichten *My Taste*, *Most Popular* und *All in Area* keine falschen Icons dargestellt werden.

Beschreibung	Aktion	Erwartetes Verhalten	Pass/Fail	Bemerkungen
Restaurants auf Karte anzeigen (Kontext "My Taste").	Auf der Map-Ansicht den Reiter *MyTaste* klicken.	Auf der Karte werden nur runde Icons mit Nummern von 1-5 und grüne Icons angezeigt. Keine schwarzen!	✓	
Restaurants auf Karte anzeigen (Kontext "Most Popular").	Auf der Map-Ansicht den Reiter *MostPopular* klicken.	Auf der Karte werden nur eckige Icons mit Nummern grösser 3 angezeigt.	✓	
Restaurants auf Karte anzeigen (Kontext "All in Area").	Auf der Map-Ansicht den Reiter *AllInArea* klicken.	Es werden alle Resultate angezeigt.	✓	

6.3.c Listenansicht der Resultate

Mittels diesen Tests wurde das korrekte Verhalten der Listenansicht überprüft.

Beschreibung	Aktion	Erwartetes Verhalten	Pass/Fail	Bemerkungen
Von der Map-Ansicht in die Listen-Ansicht wechseln.	Auf der Map-Ansicht den List-Button oben rechts klicken.	Die Map dreht sich und auf der Rückseite erscheint eine Liste. Der Button oben rechts wechselt in einen Map-Button.	✓	
Überprüfen, ob nur Restaurants in der Liste erscheinen, welche auf der Karte sichtbar sind.	Auf der Map heranzoom, damit nur wenige Restaurants im Kartenausschnitt liegen. Liste anzeigen.	In die Listenansicht werden nur jene Restaurants angezeigt, welche im Kartenausschnitt liegen.	(✓)	Die Funktion, welche feststellt, ob ein Restaurant im Kartenausschnitt liegt ist eine Vereinfachung und nicht 100% genau.

Listenansicht im Kontext *MyTaste*.	Auf der Map-Ansicht den Reiter *MyTaste* klicken.	In der Liste werden Restaurants mit runden Icons mit Nummern von 1-5 und grünen Icons angezeigt. Keine schwarzen!	✓	
Listenansicht im Kontext MostPopular.	Auf der Map-Ansicht den Reiter *MyTaste* klicken.		✓	
			✓	

6.3.d Lokalität bewerten

Beschreibung	Aktion	Erwartetes Verhalten	Pass/Fail	Bemerkungen
In der Detailansicht bewerten.	Über das Popup in der Detailansicht eine Bewertung senden.	Popup schliesst sich und das Icon in der kleinen Mapansicht zeigt die Nummer der Bewertung an.	✓	
Unter *Places to Rate* bewerten.	Im Register *Rate Places* eine Lokalität unter *Places to Rate* bewerten. In das Register *Already Rated Places* wechseln.	Die Lokalität verschwindet aus der Liste. Die Zahl neben dem Titel *Already Rated Places* erhöht sich um 1. Die Lokalität befindet sich nun unter *AlreadyRatedPlaces*.	✓	

6.3.e Allgemeine Funktionen

Beschreibung	Aktion	Erwartetes Verhalten	Pass/Fail	Bemerkungen
Infobox mit Informationen zur Funktionsweise von Localina anzeigen.	Im Tab *Find Places* den Info-Button oben rechts klicken.	Infobox erscheint auf dem Display.	✓	
Restaurants in meiner Umgebung anzeigen.	Unter dem Tab *Find Places*, *Nearby* anwählen.	Die Map zeigt den aktuellen Standort und die Restaurants werden auf der Karte angezeigt. Zu Beginn befindet man sich im Kontext *MyTaste*.	✓	
Restaurants an einer bestimmten Adresse anzeigen (via Suchfeld).	Unter FindPlaces die Option *By Address* wählen und über das Suchfeld nach der gewünschten Adresse suchen.	Die Map zeigt die gewünschte Adresse an und die Restaurants werden auf der Karte angezeigt. Zu Beginn befindet man sich im Kontext *MyTaste*.	✓	
Restaurants an einer bestimmten Adresse anzeigen (Adresse aus der Such-History wählen).	Unter FindPlaces die Option *By Address* wählen und eine Adresse aus *Recent Searches* wählen.	Die zuletzt gesucht Adresse sollte unter *Recent Searches* erscheinen. Nach dem Selektieren einer Adresse sollte die Map erscheinen und die gewählte Adresse anzeigen.	✓	

6.4 Statische Code-Analyse

Der Code wurde während der Entwicklung mit dem (Clang) Static Analyser auf Fehler geprüft. Dieser wendet eine Reihe von Algorithmen an, um Fehler im Source-Code zu erkennen. Der (Clang) Static Analyzer ist seit Xcode 3.2 (Snow Leopard) in Xcode integriert.

Wenn man den Code der Localina iPhone App mit dem Static Analyzer prüft, wird folgender Hinweis angezeigt:

```
GeocodeService.m:26:6: Potential null dereference.  According to coding standards in
'Creating and Returning NSError Objects' the parameter 'error' may be null
```

Diese Warnung kann ignoriert werden, weil bei der Implementation damit gerechnet wurde, dass der Pointer des NSErrors null sein kann.

Ansonsten liefert der Static Analyser keine Hinweise auf Fehler im Code, wie z.B. Memory Leaks.

6.5 Performance Tests

Das Laufzeitverhalten der Localina iPhone App wurde mit Hilfe des Performance Tools *Leaks* getestet. Leaks *Leaks* ist ein äusserst effizientes Tool, um Memory Leaks zu eruieren. Es sucht in regelmässigen Abständen nach Objekten im Memory, zu welchen es keinen Pointer mehr hat und zeigt an, welche Objekte wo im Programmcode verloren gegangen sind. *Leaks* ist seit der Xcode Version 3 in der Entwicklungsumgebung integriert.

Sämtliche Funktionen der Localina iPhone App wurden mit Hilfe von Leaks ausführlich getestet. Es konnten dabei keine Memory Leaks festgestellt werden, welche innerhalb des Programmcodes auftreten.

Hinweis: Kurz dem Start der Applikation, zeigt *Leaks* ein Memory Leak an. Dieses tritt jedoch innerhalb der CoreGraphics-Library auf und ist nicht als kritisch zu betrachten, da das Memory nur einmal verloren geht.

6.6 Möglichkeiten der Weiterentwicklung

Im Verlauf der Entwicklungsphase von Localina haben sich nach Vergleichen mit ähnlichen sich bereits auf dem Markt befindlichen Applikationen (wie züritipp, Tellmewhere etc.) und insbesondere auch anhand des durchgeführten Usability Test diverse Verbesserungs- und Ausbaumöglichkeiten aufgezeigt. Die Umsetzung all dieser Ideen hätte jedoch den Rahmen der Arbeit gesprengt. Trotzdem möchte ich hier kurz Möglichkeiten zur Verbesserung nennen, die sich aufgezeigt haben.

Einbindung von Facebook, Twitter, Foursquare etc.

Für viele Leute ist es ein Bedürfnis, sich über Dienste wie Facebook, Twitter oder Blogs mitteilen zu können. Das ausgeprägte Mitteilungsbedürfnis der Benutzer von sozialen Netzwerken könnte sich Localina zum Vorteil nehmen, um mehr User Generated Content und Bewertungen zu bekommen. Durch die Anbindung von Localina an Dienste wie Facebook, Twitter oder Foursquare könnte Localina auch von Effekten des viralen Marketing profitieren und dadurch den eigenen Bekanntheitsgrad steigern.

Einbindung von OpenTable

Je länger je mehr haben Menschen das Bedürfnis, zeit- und ortsunabhängig zu agieren. OpenTable bietet - anstelle der klassischen, zeitgebundenen Tischreservation per Telefon - die Möglichkeit, bei den beteiligten Restaurants kostenlos und rund um die Uhr Reservationen online vorzunehmen. Die Suche reflektiert die tatsächliche Verfügbarkeit von Tischen in „Echtzeit“ und Reservierungen werden umgehend in dem selben elektronischen Reservierungsbuch festgehalten, das vom Restaurant verwendet wird. Als registriertes Mitglied können zudem Reservationen verwaltet und Begleitpersonen per E-Mail zum Essen eingeladen werden.

Einbindung von Menukarten

Einige Testbenutzer haben den Wunsch geäussert, in der Info-Ansicht der einzelnen Restaurants direkt auf die aktuelle Menukarte zugreifen zu können. Ein Blick auf die

Menukarte würde ihnen die Wahl eines passenden Restaurants zusätzlich vereinfachen und die Vorfreude auf den Besuch des Restaurants erhöhen.

Ausweitung auf Bars, Cafés

Um den Benutzern von Localina einen umfassenden Guide zur Freizeitgestaltung bieten zu können, müsste die Localina iPhone App weitere Arten von Lokalitäten wie Bars, Cafés, Clubs und Hotels ausgeweitet werden. Allerdings könnte sich Localina auch bewusst als personalisierter Restaurant-Führer auf dem Markt platzieren und würde sich dadurch klarer vom Mitkonkurrent Tellmewhere abgrenzen.

Suche nach Kriterien

Einige Testbenutzer würden ihre Restaurant-Suche gerne nach Kriterien wie Preisklasse, Art der Küche, Ambiente etc. einschränken lassen können. Bisher wurde auf eine solche Funktion bewusst verzichtet, um nicht vom Hauptzweck von Localina abzulenken, nämlich auf den persönlichen Geschmack zugeschnittene Empfehlungen zu erhalten.

Aktivierung des Kartenausschnitts in der Info-Ansicht

Anhand der durchgeführten Benutzertests hat sich gezeigt, dass einige User in der Info-Ansicht zu einem bestimmten Restaurant den kleinen Kartenausschnitt anklicken wollten. Dadurch wollten sie entweder zur Karte, auf der mehrere Restaurants angezeigt werden, zurückkehren oder nachschauen, wo sich das Restaurant genau befindet.

Verlinkung Homepage

Um über die Info-Ansicht direkt zur Homepage des ausgewählten Restaurants zu gelangen, müsste die URL noch verlinkt werden.

Änderung Pinfarbe

Obschon die Benutzer die Ziffer 5 immer als Bestes erkannten, wäre es sinnvoller, die Abstufung der Farben bei den jeweiligen Pins umgekehrt als bisher anzubringen. Das heisst, der Button mit der Ziffer 5 sollte heller sein, als derjenige mit der Ziffer 1. Bisher ist die Ziffer 1 heller als die 5.

Informations-Button in der Kartenansicht

Es wäre sinnvoller, den Button zum Aufrufen von Informationen zur Funktionsweise von Localina und insbesondere zur Bedeutung der Farbgebung der Pins direkt in die Kartenansicht einzubinden. Bis anhin befindet sich der Button zu Beginn der Suchanfrage. Gelangt der User aus der Kartenansicht zum Informations-Button, um sich wieder in Erinnerung zu rufen, was die Pinfarbe bedeutet, so muss er seine Suchanfrage erneut eingeben, um wieder zur selben Kartenansicht zurück zu kommen.

Löschen von Restaurants in Favoritenliste

Aus einem Benutzertest hat sich ergeben, dass das Löschen eines Restaurants aus der Favoritenliste sinnvoll wäre. Begründet wurde dies damit, dass einem beispielsweise bei einem Pächterwechsel das Restaurant plötzlich nicht mehr gefallen könnte.

7 Projektmanagement

7.1 Projektplan

Der Projektplan ist unter im Verzeichnis *Projektplan* auf der Abgabe-CD zu finden.

7.2 Risiken

Erläuterungen:

- WK bezeichnet die Wahrscheinlichkeit, dass das Risiko während des Projektes auftritt.
- Der Aufwand wurde in Tagen geschätzt.
- Das Risiko wird bestimmt durch WK * Aufwand

ID	Risiko	Massnahme	WK	Aufw.	Risiko
R01	**Absenzen bringen den Zeitplan durcheinander und verzögern den Projektfortschritt.**	• Reserven im Zeitplan. • Wichtige Deliverables früh erledigen. • Tasks mit hohem Risiko früh beginnen.	0.5	2	1
R02	**Datenverlust.**	• SVN für Code. • Confluence für Projektdokumentation. • Apple Time-Machine für die Sicherung der lokalen Dateien auf einer externen Harddisk.	0.1	0.25	0.025
R03	**Die Anforderungen sind in der gegebenen Zeit nicht zu erfüllen.**	• Anforderungen nach Prioritäten ordnen und Kernanforderungen priorisieren. • Saubere Dokumentation für Weiterentwicklung.	0.1	3	0.3

ID	Risiko	Massnahme	WK	Aufw.	Risiko
R04	**Applikation lässt sich auf Zielsystem nicht installieren.**	• Installation dokumentieren und Systemanforderungen definieren. • Applikation vor Abgabe auf Systemumgebung des Betreuers installieren.	0.1	0.5	0.05
R05	**Memory Leaks generieren unerwartet hohen Aufwand für Refactorings.**	• Tools zur Erkennung von Memory Leaks evaluieren und während der gesamten Refactoring Phase einsetzen. • Von Beginn weg sauberen Code schreiben.	0.5	4	2
R06	**SOAP Service nicht erreichbar.**	• Fehlerbehandlung. • Kommunikation mit Astina. • Evtl. Mock-Service.	0.1	(3)	0.3
R07	**2. Usability Test zeigt auf, dass die Applikation nicht funktioniert.**	Nötige Anpassungen falls möglich in Refactoringphase vornehmen, ansonsten dokumentieren.	0.2	5	1

7.3 Projektorganisation

7.3.a Annahmen und Einschränkungen

Der Aufwand für das Softwareprojekt wird auf zirka 25 Stunden pro Woche geschätzt. Weil ich neben dem Projekt durch meine Arbeit bei Astina AG zeitlich stark ausgelastet bin, sind die zeitlichen Ressourcen pro Woche auf maximal 32 Stunden begrenzt. Vom 15. März bis 1. April bin ich zudem im Militärdienst und kann daher während dieser Zeit nur maximal zwei Tage am Softwareprojekt arbeiten. Dies wurde im Projektplan entsprechend berücksichtigt.

7.3.b Externe Schnittstellen

Name	Rolle	Firma/ Organisation	Kürzel	Kontakt
Prof. Dr. Markus Stolze	Betreuer HSR	HSR, IFS	mstolze	mstolze@hsr.ch
Marc Gasser	Ansprechpartner Auftraggeber	Astina AG	mgasser	mgasser@astina.ch
Prof. Dr. Lothar Müller	Gegenleser	HSR	lmueller	lmueller@hsr.ch
Michael Graf	Assistent IFS, Experte für Inhaltliche Fragen und Reviews	HSR, IFS	mgraf	mgraf@hsr.ch

7.4 Entwicklungstools

Zweck	Applikation	Version
IDE, statische Code-Analyse, GUI-Builder	Xcode / Interface Builder	3.2.2
VCM/SCM	SVN	1.6.9
Dynamische Code-Analyse (Leak-Tracking)	Instruments	2.1

7.5 Tools für das Projektmanagement

Zweck	Applikation	Beschreibung
Projekt-WIKI	Confluence	Im Confluence wurden die
Taskmanagement	Jira	Jira wurde für die Erfassung der Tasks und für die Zeiterfassung verwendet.
Projektplan	Google Docs	Für den Projektplan wurde Google Docs verwendet.

III Software-Dokumentation

1 Installation

1.1 Systemanforderungen

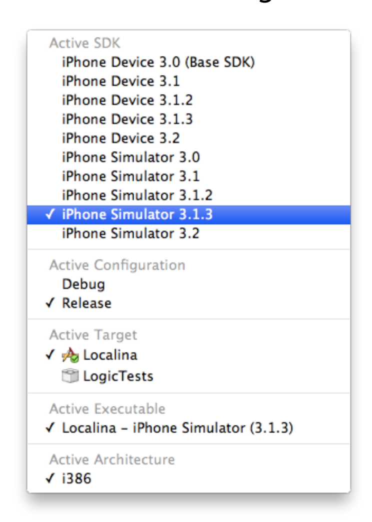
Die Localina iPhone App verwendet Funktionen, welche nur mit der Version 3 der iPhone SDK funktionieren. Deshalb benötigt man zur Installation Xcode der Version 3 oder höher.

1.2 Installation im iPhone Simulator

(1) Kopieren Sie das Verzeichnis *Localina2.0* auf der Abgabe-CD in Ihren Workspace.

(2) Öffnen Sie die Datei Localina.xcodeproj im Verzeichnis *Localina2.0*, um das Projekt in Xcode zu öffnen.

(3) Wählen Sie die folgenden Einstellungen:



(4) Wählen Sie “Build and Run”.

1.3 Hinweis

Während dem Build-Prozess wird folgende Warnungen angezeigt:

```
Libraries/TouchXML/CXMLDocument.m:229:0
/Users/smorger/workspaces/iphone/Localina/Classes/Libraries/TouchXML/CXMLDocument.m
:229: warning: no '-_XMLStringWithOptions:appendingToString:' method found
```

Diese Warnung kann ignoriert werden, weil sie innerhalb einer externen Bibliothek (TouchXML) auftritt.

2 Benutzerhandbuch

Ein kurzes Manual zur Localina iPhone Applikation findet sich auf der Abgabe-CD

3 Projektmonitoring

3.1 Soll-Ist-Zeit-Vergleich

3.1.a Nach Minimalanforderungen

3.1.b Backlog

ID	User Story	Business Value	Erfüllt
US02	**Als Benutzer möchte ich auf den ersten Blick sehen, welche Funktionen mir die Applikation bietet.**	B	✓
US03	**Als Benutzer möchte ich die Lokalitäten auf der Karte betrachten können.**	A	✓
US04	**Als Benutzer möchte ich die Lokalitäten in einer Liste betrachten können.**	B	✓
US05	**Als Benutzer möchte ich eine Lokalität in der Nähe finden.**	A	✓
US07	**Als Benutzer möchte ich eine Lokalität im Umkreis eines anderen Standorts als meines aktuellen finden.**	A	✓
US08-1	**Als Benutzer möchte ich sehen, welche Lokalitäten meinem persönlichen Geschmack entsprechen.**	A	✓
US08-2	**Als Benutzer möchte ich sehen, welche Lokalitäten meinem persönlichen Geschmack entsprechen.**	B	✓
US09	**Als Benutzer möchte ich sehen, wie beliebt eine Lokalität bei den anderen Benutzern ist.**	B	✓
US10	**Als Benutzer möchte ich Informationen zu einer Lokalität ansehen können.**	A	✓

US13	**Die Benutzer sollen regelmässig daran erinnert werden, Lokalitäten zu bewerten.**	B	✓
US14	**Als Benutzer möchte ich eine Lokalität direkt in der Detailansicht bewerten können.**	B	✓
US15	**Als Benutzer möchte ich sehen, wieviele Lokalitäten ich bewerten muss, um gute Ergebnisse zu erhalten.**	B	✓
US19	**Ladeanzeige, wenn Daten über das Internet geladen werden.**	A	✓

3.2 Code Metriken (Vorher-Nachher-Vergleich)

Für die Bestimmung der Code Metriken wurde das CLOC verwendet (http://cloc.sourceforge.net).

Metrik	Stand erster Prototyp	Aktueller Stand
Number of Files	40	101
Number of Lines (Comments)	872	1299
Number of Lines (Code)	4346	6389

CoverStory (http://code.google.com/p/coverstory) ist ein Tool zur Bestimmung der Test Coverage in Xcode. Leider verursachte das Tool einen Compiler-Fehler, weshalb ich die Test Coverage nicht bestimmen konnte.

3.3 Protokolle der wöchentlichen Sitzungen

Die Protokolle der wöchentlichen Sitzungen befinden sich im Verzeichnis *Protokolle* auf der Abgabe-CD.

IV Anhang

A Inhalt der CD

- Abstract.pdf
- Aufgabenstellung.pdf
- **Literature** (Dokumente, welche als Informationsquelle für die Arbeit gedient haben)
- Localina2.0 (Xcode-Projekt mit allen Quelldateien)
- ManagementSummary (Separate Management Summary der Bachelorarbeit)
- **Mockups** (Die originalen Mockups, welche zu Beginn des Redesigns entstanden sind)
  - Mockup-v1.pdf
  - Mockup-v.2.pdf
  - Mockup-v2.1.pdf
  - Mockup-v2.2.pdf
- **Originaldateien**
  - **Diagramme**
    - **UML**
  - **Dokumente**
  - **Illustrationen**
- Plagiats-Erklaerung.pdf
- Poster.pdf
- **Projektplan**
- **Protokolle**
  - WoechentlicheSitzungen.pdf
- **Sequenzdiagramme**
- Urheber-_und_Nutzungsrechte.pdf
- **UsabilityTests** (Dokumentation der Usability Tests)

- **Ausgangslage**
- **Mockups**
- **Release1**

B Glossar und Abkürzungsverzeichnis

Category Kategorien in Objective-C sind Erweiterungen einer Klassen um zusätzliche Methoden.

Certainty Sicherheit, mit welcher das Empfehlungssystem eine Prediction machen kann.

Cold-Start Problem Bekanntes Problem von Empfehlungsalgorithmen, welche den Collaborative-Filtering Approach verwenden. Weil diese Vergleiche zwischen Benutzerpräferenzen anstellen, können sie bei einer geringen Benutzerbasis keine zufriedenstellenden Ergebnisse liefern (vgl. **Sparse-Data Problem**).

LBS (Location Based Services) Dienste, welche von mobilen Geräten verwendet werden, um positionsabhängige Daten des Benutzers zu ermitteln.

Localina Application Localina Application ist für die Verwaltung der Location-Daten und der Benutzerprofile verwantwortlich und stellt über einen Webservice verschiedene Funktionen für Localina Clients wie die Localina iPhone App zur Verfügung.

Location Bezeichnet im Kontext der Localina iPhone App den Benutzerstandort. Bei Localina steht der Begriff Location für eine Lokalität.

Lokalität Sammelbegriff für Lokalitäten wie Bar, Clubs, Hotel, Restaurants, Cafés.

Place Bezeichnet im Kontext der Localina iPhone App eine Lokalität, weil der von Localina verwendete Begriff *Location* zu Verwechslung mit dem Benutzerstandort (engl. User location) führt.

Popularity Durchschnittliches Rating aller Benutzer der Lokalität.

Prediction Vorhersage des Systems über die Präferenz des Benutzers.

Rating Bewertung einer Lokalität durch den User.

SOAP Simple Object Access Protocol, Netzwerkprotokoll zum Austausch von Daten zwischen Systemen.

Sparse-Data Problem Das Sparse Data Problem tritt dann auf, wenn der Empfehlungsalgorithmus zuwenig umfangreiche Daten zur Verfügung hat, um qualitative guten Empfehlungen liefern zu können. (vgl. **Cold-Start Problem**)

Suchkontext Filter für die Suchresultate. Der Suchkontext kann über die Register *My Taste*, *Popular Places* und *All Places* bestimmt werden.

Think Aloud Methode bei Usability Test, wo die Testperson ihre Gedanken und Überlegungen laut mitteilt, um den Testbeobachtern mehr Informationen preiszugeben.

User Benutzer der Localina iPhone App.

C Literatur- und Quellenverzeichnis

Aus folgenden Quellen konnte ich nützliche Informationen entnehmen:

[1] Craig Larman, Applying UML and Patterns, Prentice Hall, 1998

[2] Core Data Tutorial for iPhone OS, Apple Inc., 2009

[3] David Barnard et. Al., iPhone User Interface Design Projects, Apress, 2009

[4] AdMob Mobile Metrics Report, admob, 2010